

C

Calculus of Variations

► [Variational Analysis](#)

Calibration

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Related Concepts

► [Camera Calibration](#); ► [Hand-Eye Calibration](#)

Definition

According to McGraw-Hill Encyclopedia of Science and Technology [1], calibration is the process of determining the performance parameters of an artifact, instrument, or system by comparing it with measurement standards. Adjustment may be a part of a calibration, but not necessarily. A calibration assures that a device or system will produce results which meet or exceed some defined criteria with a specified degree of confidence.

Background

In computer vision, there are multiple calibration problems. The most fundamental one is the camera

calibration, which determines the intrinsic and extrinsic parameters of a camera. It is the first step toward 3D computer vision. Other problems include hand-eye calibration, color calibration, and photometric calibration.

As stated in [1], two important measurement concepts related to calibration are precision and accuracy. Precision refers to the minimum discernible change in the parameter being measured, while accuracy refers to the actual amount of error that exists in a calibration. All measurement processes used for calibration are subject to various sources of error. It is common practice to classify them as random or systematic errors. When a measurement is repeated many times, the results will exhibit random statistical fluctuations which may or may not be significant. Systematic errors are offsets from the true value of a parameter and, if they are known, corrections are generally applied, eliminating their effect on the calibration. If they are not known, they can have an adverse effect on the accuracy of the calibration. High-accuracy calibrations are usually accompanied by an analysis of the sources of error and a statement of the uncertainty of the calibration. Uncertainty indicates how much the accuracy of a calibration could be degraded as a result of the combined errors.

References

1. Parker SP (ed) (1982) McGraw-Hill encyclopedia of science and technology, 5th edn. McGraw-Hill, New York. <http://www.answers.com/topic/calibration>

Calibration of a Non-single Viewpoint System

Peter Sturm

INRIA Grenoble Rhône-Alpes, St Ismier Cedex,
France

Synonyms

Non-central camera calibration

Related Concepts

► Camera Calibration; ► Center of Projection

Definition

A non-single viewpoint system refers to a camera for which the light rays that enter the camera and contribute to the image produced by the camera, do not pass through a single point. The analogous definition holds for *models* for non-single viewpoint systems. Hence, a non-single viewpoint camera or model does not possess a single center of projection. Nevertheless, a non-single viewpoint model (NSVM), like any other camera model such as the pinhole model, enables to project points and other geometric primitives, into the image and to back-project image points or other image primitives, to 3D. Calibration of a non-single viewpoint model consists of a process that allows to compute the parameters of the model.

Background

There exist a large variety of camera technologies (“regular” cameras, catadioptric cameras, fish-eye cameras, etc.) and camera models designed for these technologies. Often, technologies are developed in order to accommodate a desired model, such as for example, to provide a uniform spatial resolution.

Most cameras used in computer vision and other areas, can be well modeled by so-called single viewpoint, or central, camera models. These usually

model the 3D-to-2D mapping carried out by a camera, via lines of sight (or, camera rays) that all pass through a single point (the center of projection or optical center), and a mapping from these lines of sight to the image points where they hit the image plane.

Some camera types, especially some cameras having a wide field of view, but not only these, cannot be modeled very well using a single viewpoint model. This may be the case because a camera was designed to possess lines of sight that do not pass through a single point. This is, for example, the case for catadioptric cameras where the mirror surface is of conical shape: even if the camera looking at the mirror is positioned on the mirror’s axis, the lines of sight of the system do not converge to a single point, rather there exists a *viewpoint locus*. Another example is a single-lens stereo system consisting of a pinhole camera and two planar mirrors, such that the obtained images represent two perspective images acquired from two different effective viewpoints.

A camera may also be unintentionally of the non-single viewpoint type, for example, catadioptric cameras that were designed to have a single viewpoint but that due to a bad alignment of the camera and the mirror of the system, lose the single viewpoint property. Another example are fish-eye cameras; fish-eye optics are complex, and in principle, one can probably consider them as non-single viewpoint systems. However, in this and the previous example, it is not clear without further investigation of the actual system under consideration, if a single viewpoint model or an NSVM is better suited. This indeed depends on “how much” the system deviates from having a single viewpoint, how close the scene is in a typical application, how much image resolution is available, and so forth. This issue is further discussed in [1].

In the following, it is supposed that calibration is performed by acquiring one or several images of a calibration object, whose geometry is known and whose characteristic features (for simplicity, points shall be considered) can be extracted and identified in images.

Theory

There are different types of NSVM’s. One usually distinguishes parametric from non-parametric such models. For example, for non-single viewpoint



catadioptric systems, if the shape of the system's mirror is known or is known to belong to a parametric family of shapes, then the entire system can be described by few parameters: intrinsic parameters of the camera looking at the mirror, relative pose of mirror and camera, and possibly, shape parameters for the mirror. If such a parametric model is considered, calibration is, conceptually speaking, analogous to that of pinhole cameras. The main difference to calibration of pinhole cameras usually concerns the initialization process that allows to compute initial estimates of the camera parameters. Other than that, one may, in general, formulate calibration by a bundle adjustment type optimization of camera parameters, by minimizing, for example, the reprojection error, that is, a measure related to the distance between the predicted projections of points of the calibration object and those extracted in the images. Examples of parametric NSVM's are the so-called two-plane and GLC models [2–5], where lines of sight are parameterized by linear or higher-order transformations applied to points in two basis planes, other similar models where lines of sight are parameterized by linear transformations [6–9], models for pushbroom and X-slit cameras [10–13], and others [14].

A different concept consists in using non-parametric models to calibrate cameras. An example is the raxel model introduced by Grossberg and Nayar [15]. It essentially associates, to each pixel, a ray in 3D supporting the line of sight, and possibly properties such as its own radiometric response function. Importantly, one may use such a model without making any assumption about a parametric relationship between the position of pixels and the position and orientation of the associated lines of sight. Rather, one may store the coordinates of the lines of sight of all pixels, in a look-up table. Simplified versions of this model (without considering optical properties for individual pixels) have been used in several similar calibration approaches, for example, [15–18].

The principle of these approaches is thus to compute, for every camera pixel, a line of sight in 3D. To do so, at least two images of a calibration object are required. The simplest scenario considers the case where the calibration object is displaced by some known motion, between the image acquisitions. For each image, one has to estimate correspondences between camera pixels and points on the calibration

object. One way of achieving such dense correspondences is to use structured light principles, for instance, to use as calibration object a computer screen and to display a series of black and white patterns on it that encodes each pixel of the screen by a unique sequence of black and white intensities (e.g., Grey codes). Once correspondences of camera pixels and points of the calibration object (pixels of the computer screen in the above example) are known, the lines of sight can be computed by simply fitting straight 3D lines to the matched points on the calibration object. To do so, the latter must be expressed in the same 3D coordinate system, which is possible since it was assumed above that the motion of the calibration object between different acquisitions, is known. This approach was proposed independently by different researchers [15–17].

The above approach requires a minimum of two images, for different positions of the calibration object, and knowledge of the object's displacements. An extension to the case of unknown displacements was proposed in [18]. That approach requires at least three images; from matches of camera pixels and points on the calibration object, it first recovers the displacements of the object using an analysis of this scenario's multi-view geometry, and then computes lines of sight as above. Other approaches following this line are [19, 20].

The above approaches compute, for each camera pixel, an individual line of sight. If one assumes that the relation between pixels and lines of sight is particular, for instance, radially symmetric about an optical axis, then alternative solutions become possible. Such a possibility is to use a non-parametric representation of the distortion or undistortion function of a camera, that is, a function that maps viewing angles (angles between lines of sight and the optical axis) to distances in the image, between image points and the principal point or a distortion center. This can be done for both, single viewpoint and non-single viewpoint models. In the former case, it is assumed that all lines of sight pass through a single center of projection, whereas in the latter case, the model usually includes a mapping from viewing angles to the position of the intersection between lines of sight and the optical axis. Approaches of the latter type include [21, 22]. Besides making and using the assumption that the camera is radially symmetric, these calibration approaches resemble those explained above.

Application

All approaches described above, be they parametric or non-parametric, allow to perform 3D-to-2D projection and/or 2D-to-3D back projection, the latter meaning the mapping from an image point to the associated line of sight. By definition, the parametric models give analytical expressions to perform these operations. As for non-parametric ones, projection and back projection usually imply some interpolation and, possibly, a search. For instance, if a non-parametric model consists of a look-up table that gives, for each pixel, its line of sight, then back-projection of an image point with non-integer coordinates requires interpolation, whereas projection of a 3D point requires the search of the closest line(s) of sight in the look-up table and again an interpolation stage.

Other than these particular aspects, NSVM's can be used for many structure-from-motion computations completely analogously to other camera models, in particular, the pinhole model. Among the essential building blocks of structure from motion, there are pose estimation, motion estimation, and 3D point triangulation for calibrated cameras. As for pose and motion estimation (and other tasks), one usually requires two types of methods in an application: so-called minimal methods, which perform the estimation task from the minimum required number of point matches and which can be efficiently embedded in robust estimation schemes such as RANSAC, and non-linear optimization methods that refine initial estimates obtained from minimal methods. Minimal methods for pose [23–25] and motion estimation [26] with NSVM's are formulated perfectly analogously to those for the pinhole model, although their algebraic complexity is higher. All that is required by these methods from the NSVM is to compute lines of sight of interest points that are extracted and matched to another image (for motion estimation) or to a reference object (for pose estimation). As for the non-linear optimization stage, the minimization of the reprojection errors requires 3D-to-2D projections to be carried out, which, as explained above, may require search and interpolation, in which case the computation of the cost function's derivatives may have to rely on numerical differentiation. Other than that, there is no major conceptual difference compared to pose/motion estimation with pinhole cameras.

Another essential structure-from-motion task is 3D point triangulation. Here again, suboptimal methods work with lines of sight computed by the camera model for interest points in the images, and optimal methods perform the non-linear optimization of reprojection errors, where the same considerations hold as above for pose and motion estimation.

References

1. 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
2. Chen NY (1979) Visually estimating workpiece pose in a robot hand using the feature points method. PhD thesis, University of Rhode Island, Kingston
3. Chen NY, Birk J, Kelley R (1980) Estimating workpiece pose using the feature points method. *IEEE Trans Autom Cont* 25(6):1027–1041
4. Martins H, Birk J, Kelley R (1981) Camera models based on data from two calibration planes. *Comput Graph Image Process* 17:173–180
5. Yu J, McMillan L (2004) General linear cameras. Proceedings of the 8th European conference on computer vision (ECCV), Prague, Czech Republic. pp 14–27
6. Pajdla T (2002) Stereo with oblique cameras. *Int J Comput Vis* 47(1–3):161–170
7. Ponce J (2009) What is a camera? Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Miami, USA
8. Seitz S, Kim J (2002) The space of all stereo images. *Int J Comput Vis* 48(1):21–38
9. Batog G, Gojac X, Ponce J (2010) Admissible linear map models of linear cameras. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, USA
10. Gupta R, Hartley R (1997) Linear pushbroom cameras. *IEEE Trans Pattern Anal Mach Intell* 19(9):963–975
11. Pajdla T (2002) Geometry of two-slit camera. Technical Report CTU-CMP-2002-02, Center for Machine Perception, Czech Technical University, Prague
12. Zomet A, Feldman D, Peleg S, Weinshall D (2003) Mosaicing new views: the crossed-slit projection. *IEEE Trans Pattern Anal Mach Intell* 25(6):741–754
13. Feldman D, Pajdla T, Weinshall D (2003) On the epipolar geometry of the crossed-slits projection. Proceedings of the 9th IEEE international conference on computer vision, Nice, France. pp 988–995
14. Gennery D (2006) Generalized camera calibration including fish-eye lenses. *Int J Comput Vis* 68(3):239–266
15. Grossberg M, Nayar S (2005) The raxel imaging model and ray-based calibration. *Int J Comput Vis* 61(2):119–137
16. Greban K, Thorpe C, Kanade T (1988) Geometric camera calibration using systems of linear equations. Proceedings of the IEEE international conference on robotics and automation, Philadelphia, Pennsylvania, USA. pp 562–567



17. Champleboux G, Lavallée S, Sautot P, Cinquin P (1992) Accurate calibration of cameras and range imaging sensors: the NPBS method. Proceedings of the IEEE international conference on robotics and automation, Nice, France. pp 1552–1558
18. Sturm P, Ramalingam S (2004) A generic concept for camera calibration. Proceedings of the 8th European conference on computer vision (ECCV), Prague, Czech Republic. pp 1–13
19. Ramalingam S, Sturm P, Lodha S (2005) Towards complete generic camera calibration. Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Diego, USA, vol 1. pp 1093–1098
20. Dunne A, Mallon J, Whelan P (2010) Efficient generic calibration method for general cameras with single centre of projection. Comput Vis Image Underst 114(2):220–233
21. Tardif JP, Sturm P, Trudeau M, Roy S (2009) Calibration of cameras with radially symmetric distortion. IEEE Trans Pattern Anal Mach Intell 31(9):1552–1566
22. Ying X, Hu Z (2004) Distortion correction of fisheye lenses using a non-parametric imaging model. Proceedings of the Asian conference on computer vision, Jeju Island, Korea. pp 527–532
23. Chen CS, Chang WY (2004) On pose recovery for generalized visual sensors. IEEE Trans Pattern Anal Mach Intell 26(7):848–861
24. Ramalingam S, Lodha S, Sturm P (2004) A generic structure-from-motion algorithm for cross-camera scenarios. Proceedings of the 5th workshop on omnidirectional vision, camera networks and non-classical cameras, Prague, Czech Republic. pp 175–186
25. Nistér D, Stewénius H (2007) A minimal solution to the generalised 3-point pose problem. J Math Imaging Vis 27(1):67–79
26. Stewénius H, Nistér D, Oskarsson M, Åström K (2005) Solutions to minimal generalized relative pose problems. Proceedings of the 6th workshop on omnidirectional vision, camera networks and non-classical cameras, Beijing, China

Calibration of Multi-camera Setups

Jun-Sik Kim
Korea Institute of Science and Technology, Seoul,
Republic of Korea

Synonyms

Multi-camera calibration

Related Concepts

► Camera Calibration; ► Camera Parameters (Intrinsic, Extrinsic)

Definition

Calibration of multi-camera setups is a process to estimate parameters of cameras which are fixed in a setup. It usually refers to the process to find relative poses of the cameras in a single coordinate system under the assumption of known intrinsic camera parameters.

Background

Many computer vision methods including 3D reconstruction from stereo cameras utilize the multiple cameras in a system, assuming that the relative poses of cameras in a single coordinate system is already known. While the camera calibration using a planar pattern [1] simplifies calibration process for intrinsic and extrinsic parameters of each camera, estimating camera poses in *a fixed global coordinate system* is still required.

The term “multi-camera setup” includes many different camera configurations such as a stereo, inward-looking cameras, outward-looking cameras, or camera sensor networks. Because each camera setup has different viewpoint and field of view (FOV) configuration, one single calibration method is not able to deal with all the multi-camera setups. Depending on the camera configuration, different calibration approach should be considered.

Theory

A projection matrix \mathbf{P}_i of a camera i in a multi-camera setup is given as

$$\mathbf{P}_i = \mathbf{K}_i [\mathbf{R}_i \ \mathbf{T}_i]. \quad (1)$$

The matrices \mathbf{R}_i and \mathbf{T}_i represent the pose of the camera i in a *predetermined* fixed coordinate system. More specifically, the matrices express a transformation between the fixed global coordinate system and the local camera coordinate system. The goal of the multi-camera calibration is to estimate the matrices \mathbf{R}_i and \mathbf{T}_i for all i in the system. The camera matrix \mathbf{K}_i represents intrinsic parameters of the camera i , which can be assumed to be known by calibrating the

intrinsic parameters of each camera independently in advance.

Multi-camera systems can be categorized into three configurations: inward-looking cameras, outward-looking cameras and large camera networks.

Inward-looking cameras The case that all the cameras in the system have a FOV shared. A stereo camera is considered *inward-looking* because the two cameras should see the same scene.

Outward-looking cameras The case that all the cameras in the system do not share their FOV. Because no FOVs are overlapped, it is usually called *non-overlapping cameras*.

Camera networks The case that some cameras share their FOVs but there are no common FOV for all cameras. Distributed cameras are usually in this category. Most likely, nearby cameras have a common FOV, farther cameras can not see it.

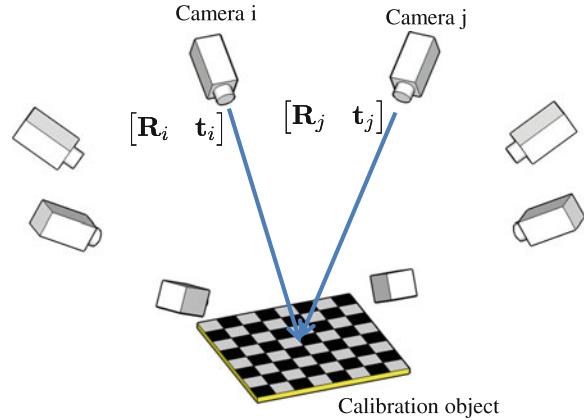
Each camera configuration has different constraints used in the multi-camera calibration, and the resulting calibration method becomes different to each other.

Inward-Looking Cameras

When all the cameras in the multi-camera setup have a common FOV, the multi-camera calibration is relatively simple. A calibration object is placed in the common FOV as shown in Fig. 1 so that each camera can see it, and the pose of the cameras with respect to the object coordinate system is estimated by using conventional pose estimation methods [2–4]. In this case, the common coordinate system of the multi-camera configuration is set to be the coordinate system of the calibration object.

For pose estimation of each camera, a planar pattern is preferable because it provides better visibility for all cameras. Note that, however, it is not limited to a planar pattern when the object visibility from every camera is ensured.

Sinha et al. [5] present an automatic calibration method using object silhouettes. In this method, epipolar geometry between cameras is estimated from dynamic silhouettes and projective structure is recovered. Following self-calibration completes the Euclidean reconstruction. This aims especially for shape-from-silhouette or visual hull reconstruction.



Calibration of Multi-camera Setsups, Fig. 1 Calibration of inward-looking cameras

Outward-Looking Cameras

When the FOVs of cameras in the system are not overlapped, it is impossible to place a calibration object which is observable from multiple cameras. The pose between cameras can be estimated by utilizing the fact that the transformation between cameras are fixed in motions called a rigidity constraint [6].

Assume that the coordinate systems of cameras i and j are transformed by a transformation \mathbf{R}_{ij} and \mathbf{t}_{ij} . When the camera i moves with a transformation $\Delta\mathbf{R}_i$ and $\Delta\mathbf{t}_i$, the motion of the camera j with a rotation $\Delta\mathbf{R}_j$ and a translation $\Delta\mathbf{t}_j$ is given as

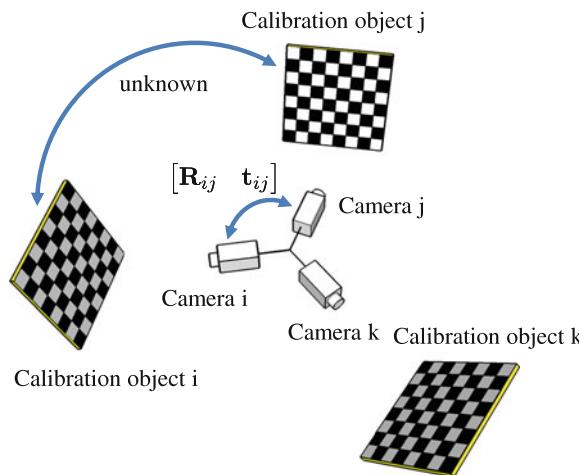
$$\begin{bmatrix} \Delta\mathbf{R}_j & \Delta\mathbf{t}_j \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{R}_i & \Delta\mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & 1 \end{bmatrix}^{-1}, \quad (2)$$

and this equation can be rewritten in a $\mathbf{AX} = \mathbf{XB}$ form on the Euclidean group as

$$\begin{bmatrix} \Delta\mathbf{R}_j & \Delta\mathbf{t}_j \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{ij} & \mathbf{t}_{ij} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{R}_i & \Delta\mathbf{t}_i \\ \mathbf{0}^\top & 1 \end{bmatrix}. \quad (3)$$

Solving the unknowns \mathbf{R}_{ij} and \mathbf{t}_{ij} on the Euclidean group is known as a *hand-eye calibration* in the robotics community [7–9]. Two or more motions of the camera rig provide enough number of constraints.

One practical problem is to estimate *motions* $\Delta\mathbf{R}$ and $\Delta\mathbf{t}$ of each camera. One stable way is to use calibration objects for each camera. First place a calibration object for each camera and take pictures while moving the camera rig. At each time frame,



Calibration of Multi-camera Setups, Fig. 2 Calibration of outward-looking cameras

the poses of each camera can be estimated by using a conventional pose estimation method. The motions of cameras are computed by calculating the difference of the poses at different time frames. Note that the geometric relation between calibration objects is not required in calculating each camera motions. The only requirement is that the calibration objects are fixed in motions. [Figure 2](#) shows the calibration objects for three non-overlapping cameras.

By solving the $\mathbf{AX} = \mathbf{XB}$ equation on the Euclidean group, the transformation between two cameras is obtained. If the multi-camera system has more than two cameras, every camera can be registered in the fixed global coordinate system by chaining the transformations pairwisely. However, the pairwise chaining of transformation does not guarantee the globally consistent registration. In addition, the transformation between cameras may be inconsistent depending on the data provided to the $\mathbf{AX} = \mathbf{XB}$ solver. Dai et al. [10] represent a rotation averaging strategy to improve the consistency of the estimated transformation, and use a global bundle adjustment [11] for final polishing. Lebraly et al. [12] more focus on an sparse implementation of the global bundle adjustment to ensure the consistency in the fixed global coordinate system.

Kumar et al. [13] present completely different approach to use a mirror so that the cameras can

observe the mirrored calibration pattern, and show successful calibration result for the ladybug camera.

Camera Networks

One general configuration is a camera network, which usually has many fixed cameras seeing in different directions. When every camera shares its FOV with any other camera in the network, relative transformations between the cameras can be estimated using calibration objects, and they are registered by chaining the transformation. In fact, this calibration process is cast to a conventional structure from motion problem. Once all the relative transformations are obtained, globally consistent localization of the cameras in the fixed coordinate system is achieved by using bundle adjustment [11]. [Figure 3](#) shows a possible placement of calibration objects in calibrating a camera network. Devarajan et al. [14] introduce a *vision graph* to describe the feature visibility between cameras, and try to optimize the graph network by belief propagation.

Baker and Aloimonos [15] present a method based on the multi-frame structure from motion algorithm, and use a rod with two LEDs at each end as a calibration object. The LEDs provides accurate and easily detectable correspondences for the precisely synchronized cameras by waiving the rod.

Svoboda et al. [16] propose a convenient and complete self-calibration method using a laser pointer including intrinsic parameter estimation. The method is based on the stratification of the transformations; at first projective reconstruction is achieved by factorization and later upgraded to Euclidean space by imposing geometric constraints such as a square pixel assumption. Their source codes are available for public use.

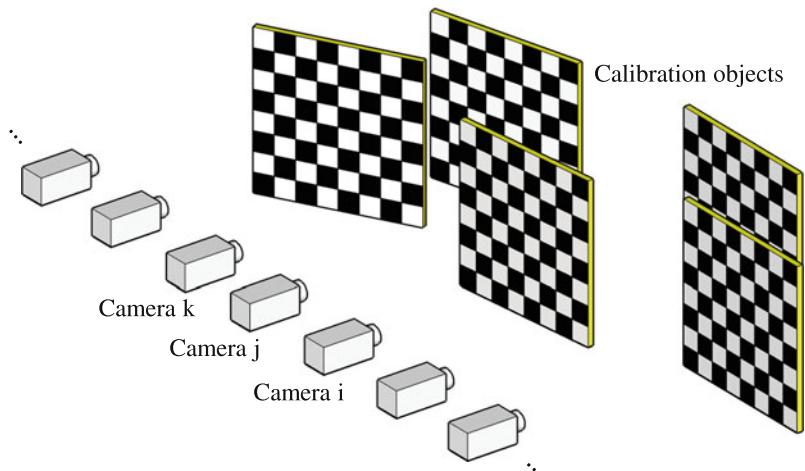
If there is no camera sharing its FOV with others, it is challenging to establish the common global coordinate system. One idea is to use a mobile robot carrying a calibration object [17]. The location of the object is estimated by the SLAM of the mobile robot. However, this is not stable enough for practical use yet.

Application

Multi-camera calibration is essential in constructing a system using multiple cameras depending on the

Calibration of Multi-camera

Setups, Fig. 3 Calibration of a camera network



applications and sensor configurations. The *inward-looking configuration* is generally used in many 3D reconstruction tasks such as stereo and visual hull reconstruction. The *outward-looking configuration* is useful to enlarge the effective FOV of the system, and especially for structure from motion applications. The most general *camera network* has diverse applications such as 3D reconstruction, surveillance, environmental monitoring and so on. Note that the camera network includes the inward-looking configuration of cameras.

References

- Zhang Z (2000) A flexible new technique for camera calibration. *IEEE Trans Pattern Anal Mach Intell* 22(11): 1330–1334
- Dementhon DF, Davis LS (1995) Model-based object pose in 25 lines of code. *Int J Comput Vis* 15:123–141
- Quan L, Lan Z (1999) Linear n-point camera pose determination. *IEEE Trans Pattern Anal Mach Intell* 21(8): 774–780
- Ababsa F, Mallem M (2004) Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems. Proceedings of the 2004 ACM SIGGRAPH international conference on virtual reality continuum and its applications in industry. ACM, New York, pp 431–435
- Sinha S, Pollefeys M, McMillan L (2004) Camera network calibration from dynamic silhouettes. Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR), vol 1. IEEE, Silver Spring, pp I–195
- Esquivel S, Woelk F, Koch R (2007) Calibration of a multi-camera rig from non-overlapping views. Proceedings of the 29th DAGM conference on pattern recognition. Springer, Berlin/New York, pp 82–91
- Andreff N, Horaud R, Espiau B (1999) On-line hand-eye calibration. Proceedings of the second international conference on 3-D digital imaging and modeling, 1999. Ottawa, Canada, IEEE, pp 430–436
- Dornaika F, Horaud R (1998) Simultaneous robot-world and hand-eye calibration. *IEEE Trans Robot Autom* 14(4): 617–622
- Park F, Martin B (1994) Robot sensor calibration: solving $ax = xb$ on the euclidean group. *IEEE Trans Robot Autom* 10(5):717–721
- Dai Y, Trampf J, Li H, Barnes N, Hartley R (2010) Rotation averaging with application to camera-rig calibration. Asian conference on computer vision. Springer, Berlin/New York, pp 335–346
- Triggs B, McLauchlan P, Hartley R, Fitzgibbon A (2000) Bundle adjustment – modern synthesis. *Vision algorithms: theory and practice*. Springer-Verlag, London, UK, pp 298–372
- Lebraly P, Royer E, Ait-Aider O, Deymier C, Dhome M (2011) Fast calibration of embedded non-overlapping cameras. IEEE international conference on robotics and automation (ICRA). Shanghai, China, IEEE, pp 221–227
- Kumar R, Ilie A, Frahm J, Pollefeys M (2008) Simple calibration of non-overlapping cameras with a mirror. IEEE conference on computer vision and pattern recognition (CVPR). Anchorage, AK, USA, IEEE, pp 1–7
- Devarajan D, Cheng Z, Radke R (2008) Calibrating distributed camera networks. *Proc IEEE* 96(10): 1625–1639
- Baker P, Aloimonos Y (2000) Complete calibration of a multi-camera network. Proceedings of the IEEE workshop on omnidirectional vision, 2000. Hilton Head Island, SC, USA, IEEE, pp 134–141
- Svoboda T, Martinec D, Pajdla T (2005) A convenient multicamera self-calibration for virtual environments. *Presence* 14(4):407–422
- Rekleitis I, Dudek G (2005) Automated calibration of a camera sensor network. IEEE/RSJ international conference on intelligent robots and systems. Edmonton, Canada, IEEE, pp 3384–3389



Calibration of Projective Cameras

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Synonyms

Lens distortion correction

Related Concepts

► Calibration; ► Camera Calibration; ► Geometric Calibration; ► Perspective Camera

Definition

Calibration of a projective camera is the process of determining an adjustment on the camera so that, after adjustment, it follows the pinhole or perspective projection model.

Background

A projective camera follows pinhole or perspective projection, which is also known as rectilinear projection because straight lines in a scene remain straight in an image. A real camera usually uses lenses with finite aperture, especially for low-end cameras (such as WebCams) or wide-angle cameras. Lens distortion also arises from imperfect lens design and manufacturing, as well as camera assembly. A line in a scene is not seen as a line in the image. A point in 3D space, its corresponding point in image, and the camera's optical center are not collinear. The linear projective equation is sometimes not sufficient, and lens distortion has to be considered or corrected beforehand.

Theory

According to [1], there are four steps in camera projection including lens distortion:

Step 1: *Rigid transformation* from world coordinate system (X_w, Y_w, Z_w) to camera one (X, Y, Z) :

$$[X \quad Y \quad Z]^T = \mathbf{R} [X_w \quad Y_w \quad Z_w]^T + \mathbf{t}$$

Step 2: *Perspective projection* from 3D camera coordinates (X, Y, Z) to *ideal* image coordinates (x, y) under pinhole camera model:

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

where f is the effective focal length.

Step 3: *Lens distortion*:

$$\check{x} = x + \delta_x, \quad \check{y} = y + \delta_y$$

where (\check{x}, \check{y}) are the *distorted* or *true* image coordinates and (δ_x, δ_y) are distortions applied to (x, y) . Note that the lens distortion described here is different from Tsai's treatment. Here, we go from ideal to real image coordinates, similar to [2].

Step 4: *Affine transformation* from real image coordinates (\check{x}, \check{y}) to *frame buffer* (pixel) image coordinates (\check{u}, \check{v}) :

$$\check{u} = d_x^{-1} \check{x} + u_0, \quad \check{v} = d_y^{-1} \check{y} + v_0,$$

where (u_0, v_0) are coordinates of the principal point and d_x and d_y are distances between adjacent pixels in the horizontal and vertical directions, respectively.

There are two types of distortions:

Radial distortion: It is symmetric; ideal image points are distorted along radial directions from the distortion center. This is caused by imperfect lens shape.

Decentering distortion: This is usually caused by improper lens assembly; ideal image points are distorted in both radial and tangential directions.

The reader is referred to [3–6] for more details.

The distortion can be expressed as power series in radial distance $r = \sqrt{x^2 + y^2}$:

$$\begin{aligned} \delta_x &= x(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ &\quad + [p_1(r^2 + 2x^2) + 2p_2xy](1 + p_3r^2 + \dots), \end{aligned}$$

$$\delta_y = y(k_1r^2 + k_2r^4 + k_3r^6 + \dots) + [2p_1xy + p_2(r^2 + 2y^2)](1 + p_3r^2 + \dots),$$

where k_i s are coefficients of radial distortion and p_j s are coefficients of decentering distortion.

Based on the reports in the literature [1, 2, 4], it is likely that the distortion function is totally dominated by the radial components and especially dominated by the first term. It has also been found that any more elaborated modeling not only would not help (negligible when compared with sensor quantization) but also would cause numerical instability [1, 2].

Denote the ideal pixel image coordinates by $u = x/d_x$ and $v = y/d_y$. By combining Steps 3 and 4 and if only using the first two radial distortion terms, we obtain the following relationship between (\check{u}, \check{v}) and (u, v) :

$$\check{u} = u + (u - u_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2] \quad (1)$$

$$\check{v} = v + (v - v_0)[k_1(x^2 + y^2) + k_2(x^2 + y^2)^2]. \quad (2)$$

Lens distortion parameters can be determined as an integrated part of geometric calibration [7]. This can be done by observing a known 3D target [1, 2, 6], by observing a 2D planar pattern [8], by observing a linear point pattern [9], or by moving the camera through a rigid scene [10]. The nonlinearity of the integrated projection and lens distortion model does not allow for a direct calculation of all the parameters of the camera model. Camera calibration including lens distortion can be performed by minimizing the distances between the image points and their predicted positions, i.e.,

$$\min_{\mathbf{A}, \mathbf{R}, \mathbf{t}, k_1, k_2} \sum_i \|\mathbf{m}_i - \check{\mathbf{m}}(\mathbf{A}, \mathbf{R}, \mathbf{t}, k_1, k_2, \mathbf{M}_i)\|^2 \quad (3)$$

where $\check{\mathbf{m}}(\mathbf{A}, \mathbf{R}, \mathbf{t}, k_1, k_2, \mathbf{M}_i)$ is the projection of \mathbf{M}_i onto the image according to the pinhole model, followed by distortion according to Eq. (1) and Eq. (2). The minimization is performed through an iterative approach such as using the Levenberg-Marquardt method.

An alternative approach is to perform lens distortion correction as a separate process. Invariance properties under projective transformation are exploited. One is the “plumb line” constraint [4], which is based on the

fact that a line in a scene remains a line in an image. Another is the cross-ratio constraint [2], which states that, for four collinear points with known distances between each other in 3D, their corresponding image points are collinear, and their cross-ratio remains the same. Due to lens distortion, projective invariants are not preserved, and we can use the variance to compute the distortion.

References

1. Tsai RY (1987) A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf tv cameras and lenses. IEEE J Robot Autom 3(4):323–344
2. Wei G, Ma S (1994) Implicit and explicit camera calibration: theory and experiments. IEEE Trans Pattern Anal Mach Intell 16(5):469–480
3. Slama CC (ed) (1980) Manual of photogrammetry, 4th edn. American Society of Photogrammetry, Falls Church
4. Brown DC (1971) Close-range camera calibration. Photogramm Eng 37(8):855–866
5. Faig W (1975) Calibration of close-range photogrammetry systems: mathematical formulation. Photogramm Eng Remote Sens 41(12):1479–1486
6. Weng J, Cohen P, Herniou M (1992) Camera calibration with distortion models and accuracy evaluation. IEEE Trans Pattern Anal Mach Intell 14(10):965–980
7. Zhang Z (2004) Camera calibration. In: Medioni G, Kang S (eds) Emerging topics in computer vision. Prentice Hall Professional Technical Reference, Upper Saddle River, pp 4–43
8. Zhang Z (2000) A flexible new technique for camera calibration. IEEE Trans Pattern Anal Mach Intell 22(11): 1330–1334
9. Zhang Z (2004) Camera calibration with one-dimensional objects. IEEE Trans Pattern Anal Mach Intell 26(7): 892–899
10. Zhang Z (1996) On the epipolar geometry between two images with lens distortion. In: International conference on pattern recognition, Vienna, Austria, vol I, pp 407–411

Calibration of Radiometric Falloff (Vignetting)

Stephen Lin

Microsoft Research Asia, Beijing Sigma Center,
Beijing, China

Synonyms

Vignetting estimation



Related Concepts

- [Irradiance](#); ► [Radiometric Calibration](#); ► [Radiance](#);
- [Vignetting](#)

Definition

Calibration of radiometric falloff is the measurement of brightness attenuation away from the image center for a given camera, lens, and camera settings.

Background

Several mechanisms may be responsible for radiometric falloff. One is the optics of the camera, which may have a smaller effective lens opening for light incident at greater off-axis angles (i.e., irradiance toward the edges of an image). Radiometric falloff also occurs naturally due to foreshortening of the lens when viewed at increasing angles from the optical axis. A third cause is mechanical in nature, where light arriving at oblique angles is partially obstructed by camera components such as the field stop or lens rim. Digital sensors may also contribute to this falloff because of angle-dependent sensitivity to light. The profile of the radiometric falloff field varies with respect to camera, lens, and camera settings such as focal length and aperture.

Many computer vision algorithms assume that the image irradiance measured at the camera sensor is equal to the scene radiance that arrives at the camera. However, this assumption often does not hold because of radiometric falloff. It is therefore important to measure or estimate the radiometric falloff, and remove its effects from images.

Theory

Radiometric falloff, or vignetting, may be modeled as a function f that represents the proportion of image brightness I at an image position (x, y) relative to that at the image center (x_0, y_0) :

$$f(x, y) = \frac{I(x, y)}{I(x_0, y_0)}. \quad (1)$$

Because of approximate radial symmetry in the optical systems of most cameras, the radiometric falloff function may alternatively be expressed in terms of image distance r from the image center:

$$f(r) = \frac{I(r)}{I(0)}, \quad (2)$$

where $r = \sqrt{x^2 + y^2}$. The purpose of radiometric falloff calibration is to recover f , so that its inverse function f^{-1} can be applied to an image i , recorded by the same camera and camera settings, to obtain an image \tilde{i} without radiometric falloff:

$$\tilde{i}(x, y) = f^{-1}(i(x, y)). \quad (3)$$

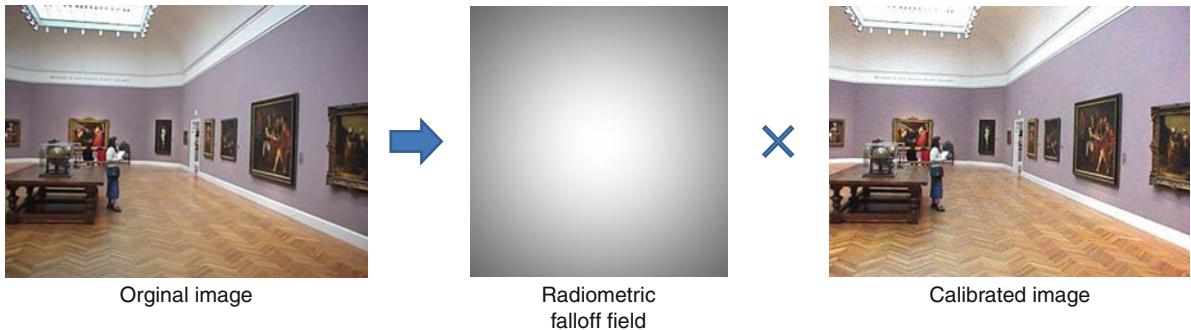
The effect of radiometric falloff calibration is illustrated in [Fig. 1](#).

Methods

A basic method for calibration of radiometric falloff is to capture a reference image consisting of a uniform radiance field [\[1–4\]](#). Since the scene itself contains no brightness variation, intensity differences in the image can be attributed solely to radiometric falloff. It must be noted that in these and other methods of falloff calibration, it is assumed that the camera response function is known.

Another approach examines image sequences with overlapping views of an arbitrary static scene [\[5–9\]](#). In overlapping image regions, corresponding points are assumed to have the same scene radiance. Differences in their intensities are therefore a result of different radiometric falloff at their respective image positions. From the positions and relative intensities among each set of corresponding points, the radiometric falloff field can be recovered without knowledge of scene content. Most of these methods are designed to recover both the radiometric falloff field and the camera response function in a joint manner [\[6–9\]](#).

The radiometric falloff field may alternatively be estimated from a single arbitrary input image. To infer the falloff field in this case, the intensity variation caused by falloff needs to be distinguished from that due to scene content. This has been done using a segmentation-based approach that identifies image regions with reliable data for falloff estimation [\[10\]](#), and by examining the effect of falloff on radial gradient distributions in the image [\[11\]](#).



Calibration of Radiometric Falloff (Vignetting), Fig. 1 Image calibrated for radiometric falloff

Application

Calibration of radiometric falloff is of importance to algorithms such as shape-from-shading and photometric stereo that infer scene properties from image irradiance values. It is also essential in applications such as image mosaicing and segmentation that require photometric consistency of the same scene point appearing in different images, or different scene points within the same image.

A measured radiometric falloff field may be used to locate the optical center of the image, since radiometric falloff generally exhibits radial symmetry. The spatial variation of light transmission may also be exploited in sensing, such as to capture high dynamic range intensity values of scene points viewed from a moving camera [12].

References

1. Sawchuk AA (1977) Real-time correction of intensity nonlinearities in imaging systems. *IEEE Trans Comput* 26(1):34–39
2. Asada N, Amano A, Baba M (1996) Photometric calibration of zoom lens systems. In: IEEE international conference on pattern recognition, Washington DC, pp 186–190
3. Kang SB, Weiss R (2000) Can we calibrate a camera using an image of a flat textureless lambertian surface? In: European conference on computer vision 2000 (ECCV), vol II. Springer, London, pp 640–653
4. Yu W (2004) Practical anti-vignetting methods for digital cameras. *IEEE Trans Consum Electron* 50:975–983
5. Jia J, Tang CK (2005) Tensor voting for image correction by global and local intensity alignment. *IEEE Trans Pattern Anal Mach Intell* 27(1):36–50
6. Litvinov A, Schechner YY (2005) Addressing radiometric nonidealities: a unified framework. In: IEEE computer

vision and pattern recognition (CVPR), IEEE Computer Society, Silver Spring, pp 52–59

7. Litvinov A, Schechner YY (2005) A radiometric framework for image mosaicing. *J Opt Soc Am A* 22:839–848
8. Goldman DB, Chen JH (2005) Vignette and exposure calibration and compensation. In: IEEE international conference on computer vision, Beijing, pp 899–906
9. Kim SJ, Pollefeys M (2008) Robust radiometric calibration and vignetting correction. *IEEE Trans Pattern Anal Mach Intell* 30(4):562–576
10. Zheng Y, Lin S, Kambhamettu C, Yu J, Kang SB (2009) Single-image vignetting correction. *IEEE Trans Pattern Anal Mach Intell* 31(12):2243–2256
11. Zheng Y, Yu J, Kang SB, Lin S, Kambhamettu C (2008) Single-image vignetting correction using radial gradient symmetry. *IEEE Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos
12. Schechner YY, Nayar SK (2003) Generalized mosaicing: high dynamic range in a wide field of view. *Int J Comput Vis* 53(3):245–267

Camera Calibration

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Related Concepts

- [Calibration](#); ► [Calibration of Projective Cameras](#);
- [Geometric Calibration](#)

Definition

Camera calibration is the process of determining certain parameters of a camera in order to fulfill desired



tasks with specified performance measures. The reader is referred to entry [►Calibration](#) for a general discussion on calibration.

Background

There are multiple camera calibration problems. The most common one, which almost becomes the synonym of camera calibration, is geometric calibration (see entry [►Geometric Calibration](#)). It consists in determining the intrinsic and extrinsic parameters of a camera.

Other camera calibration problems include:

Stereo calibration. A stereo (or stereovision) system consists of multiple cameras. Stereo calibration determines the relative geometry (rotation and translation) between cameras. The intrinsic parameters of each camera can be determined separately as in camera calibration or jointly with the relative geometry.

Photometric calibration. Photometry concerns the measurement of quantities associated with light. Photometric calibration of a camera is a process of determining a function which converts the pixel values to photometric quantities such as SI (*Système International* in French) light units. A test chart of patches with known relative luminances is usually used for photometric calibration.

Color calibration. The pixel values of a color camera depend not only on the surface reflectance but also on the illuminating source. White balance is a common color calibration task, which uses a standard test target with known reflectance to remove the influence of lighting on the scene. Another common task is to calibrate multiple seemingly identical cameras which are not due to tolerance in fabrication.

Camera Extrinsic Parameters

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Synonyms

Camera pose; Extrinsic parameters

Related Concepts

[►Camera Parameters \(Intrinsic, Extrinsic\)](#); [►Intrinsics](#)

Definition

extrinsic, short for *extrinsic parameters*, refer to the parameters not forming the essential part of a thing, which is usually a camera in computer vision. The extrinsic parameters of a camera include its pose (rotation and translation) with respect to a reference coordinate system.

See entry “[►Camera Parameters \(Intrinsic, Extrinsic\)](#)” for more details.

Camera Model

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Related Concepts

[►Camera Calibration](#); [►Camera Parameters \(Intrinsic, Extrinsic\)](#); [►Intrinsics](#); [►Perspective Camera](#)

Definition

A *camera* is a device that records lights coming from the scene and saves them in images. These images may be still photographs or moving images (videos). The *camera model* describes the mathematical relationship between the 3D coordinates of a point in the scene from which the light comes from and the 2D coordinates of its projection onto the image plane. The ideal camera model is known as the pin-hole camera model or perspective camera model, but other camera models exist such as thin and thick cameras.

Background

The term *camera* comes from the *camera obscura* (in Latin for “dark chamber”) [1]. A camera obscura is a dark room, consisting of a darkened chamber or box, into which light is admitted through a pinhole (later a

convex lens), forming an image of external objects on a surface of wall, paper, or glass. A modern camera generally consists of an enclosed hollow with an opening (aperture) at one end for light to enter and a recording or viewing surface (such as a CCD or CMOS sensor) for capturing the light on the other end. A majority of cameras have a lens positioned in front of the camera's opening to gather the incoming light and focus all or part of the image on the recording surface.

A camera may work with the light of the visible spectrum. If it records each of the red, blue, and green primary colors at each pixel, then the camera is called a color camera; if it only records the shades of black and white (the grey levels of the light intensity), the camera is called a black and white camera.

A camera may also work with the light outside of the visible spectrum, e.g., with the infrared (IR) light, and the camera is called the IR camera.

Pinhole Camera Model

A pinhole camera can be ideally modeled as the *perspective projection*. This is by far the most popularly used model in the computer vision community. The reader is referred to the entry ▶ [Perspective Camera](#) for details.

Thin and Thick Lens Camera Models

Although pinhole cameras model quite well most of the cameras we use in the computer vision community, they cannot be used physically in a real imaging system. This is for two reasons:

- An ideal pinhole, having an infinitesimal aperture, does not allow to gather enough amount of light to produce measurable image brightness (called *image irradiance*).
- Because of the wave nature of light, diffraction occurs at the edge of the pinhole and the light spread over the image [2]. As the pinhole is made smaller and smaller, a larger and larger fraction of the incoming light is deflected far from the direction of the incoming ray.

To avoid these problems, a real imaging system usually uses lenses with finite aperture. This appendix aims at having the reader know that there are other camera models available. One should choose an

appropriate model for a particular imaging device [3, Sect. 2.A.1].

For an ideal lens, which is known as the *thin lens*, all optical rays parallel to the optical axis converge to a point on the optical axis on the other side of the lens at a distance equal to the so-called *focal length* f (see Fig. 1).

The light ray through the center of the lens is undeflected; thus a thin lens produces the same projection as the pinhole. However, it gathers also a finite amount of light reflected from (or emitted by) the object (see Fig. 2). By the familiar *thin lens law*, rays from points at a distance Z are focused by the lens at a distance $-F$, and Z and $-F$ satisfy

$$\frac{1}{Z} + \frac{1}{-F} = \frac{1}{-f}, \quad (1)$$

where f is the focal length.

If we put an image plane at the distance $-F$, then points at other distances than Z are imaged as small blur circles. This can be seen by considering the cone of light rays passing through the lens with apex at the point where they are correctly focused [2]. The size of the blur circle can be determined as follows. A point at distance \hat{Z} is focused if it is imaged at a point $-\hat{F}$ from the lens (see Fig. 3), where

$$\frac{1}{\hat{Z}} + \frac{1}{-\hat{F}} = \frac{1}{-f}.$$

It gives rise to a blur circle on the image plane located at distance $-\hat{F}$. The diameter of the blur circle, e , can be computed by triangle similarity

$$\frac{e}{d} = \frac{|F - \hat{F}|}{\hat{F}},$$

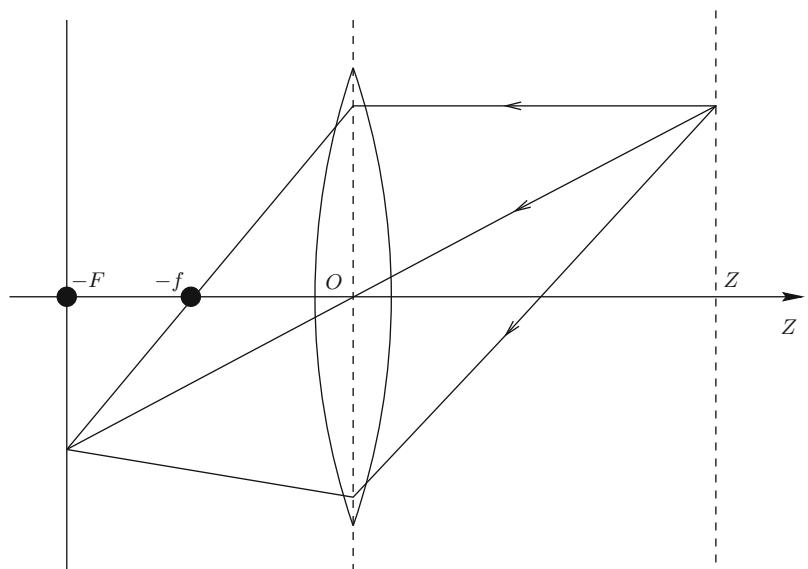
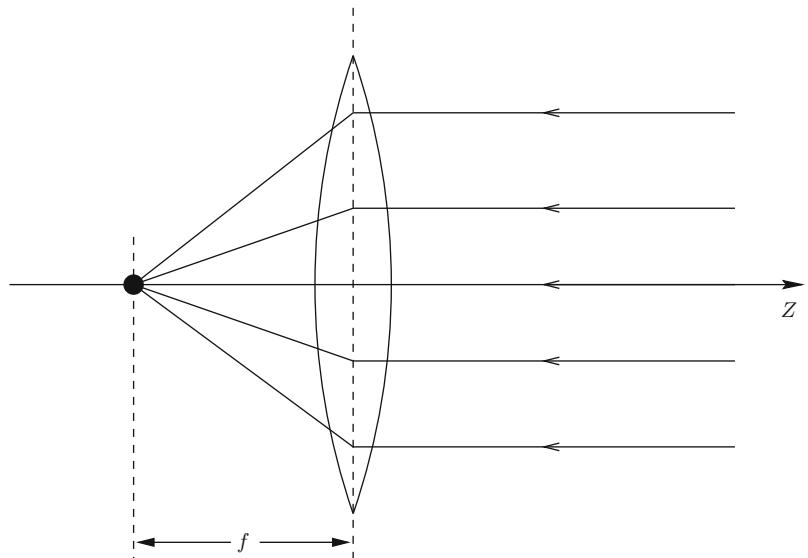
which gives

$$e = \frac{d}{\hat{F}} |F - \hat{F}| = \frac{fd}{\hat{Z}} \frac{|Z - \hat{Z}|}{Z - f},$$

where d is the diameter of the lens. If the diameter of blur circles, e , is less than the resolution of the image, then the object is well focused and its image is clean. The range of distances over which objects are focused "sufficiently well" is called the *depth of field*. It is clear

**Camera Model, Fig. 1**

Cross-sectional view of a thin lens sliced by a plane containing the optical axis. All light rays parallel to the optical axis converge to a point at a distance equal to the focal length



Camera Model, Fig. 2 A thin lens gathers light from a finite area and produces a well-focused image at a particular distance

that the larger the lens aperture d , the less the depth of field.

From (1), it is seen that for objects relatively distant from the lens (*i.e.*, $Z \gg f$), we have $F = f$. If the image plane is located at distance f from the lens, then the camera can be modeled reasonably well by the pinhole.

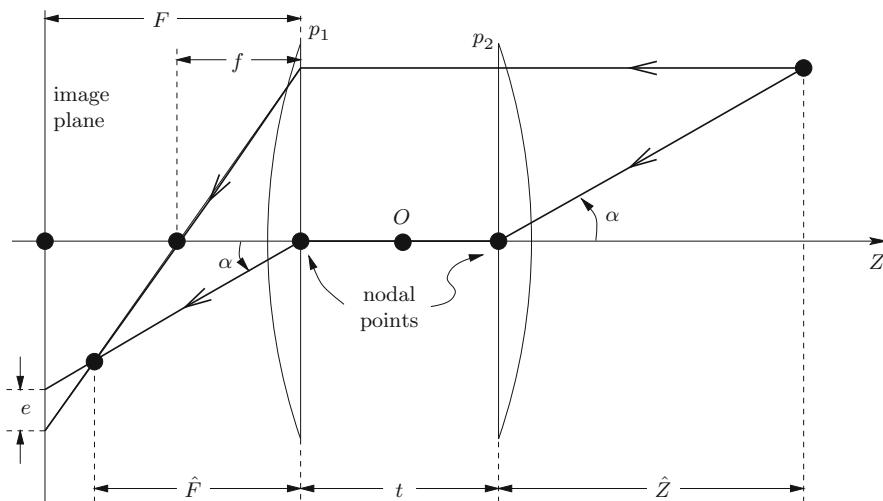
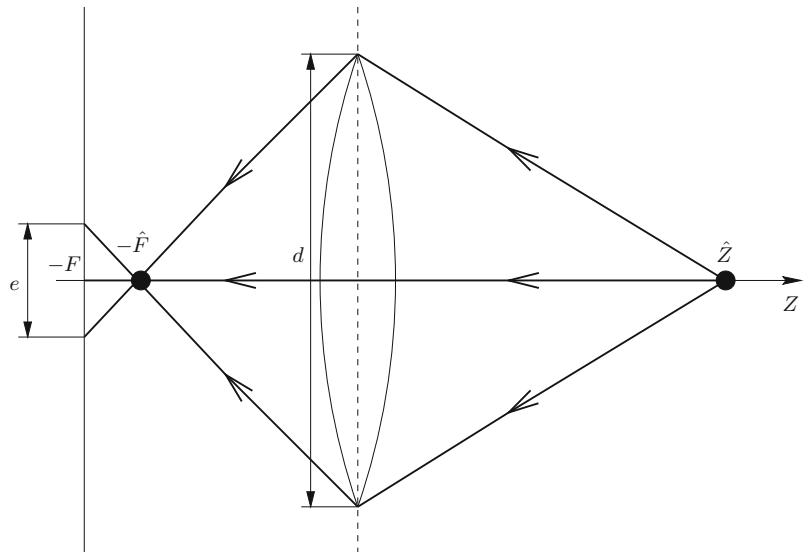
It is difficult to manufacture a perfect lens. In practice, several simple lenses are carefully assembled to make a compound lens with better properties. In an imaging device with mechanism of focus and zoom, the lenses are allowed to move. It appears difficult to

model such a device by a pinhole or thin lens. Another model, called the *thick lens*, is used by more and more researchers [4, 5].

An ideal thick lens is illustrated in Fig. 4. It is composed of two lenses, each having two opposite surfaces, one spherical and the other plane. These two planes p_1 and p_2 , called the *principal planes*, are perpendicular to the optical axis and are separated by a distance t , called the *thickness of the lens*. The principal planes intersect the optical axis at two points, called the *nodal points*. The thick lens produces the same perspective projection as the ideal thin lens,

Camera Model, Fig. 3

Focus and blur circles

**Camera Model, Fig. 4** Cross-sectional view of a thick lens sliced by a plane containing the optical axis

except for *an additional offset* equal to the lens thickness t along the optical axis. A light ray arriving at the first nodal point leaves the rear nodal point without changing direction. A thin lens can then be considered as a thick lens with $t = 0$.

It is thus clear that a thick lens can be considered as a thin lens if the object is relatively distant to the camera compared to the lens thickness (*i.e.*, $\hat{Z} \gg t$). It can be further approximated by a pinhole only when the object is well focused (*i.e.*, $F \approx \hat{F}$), and this is valid only locally.

References

1. Wikipedia (2011) History of the camera. http://en.wikipedia.org/wiki/History_of_the_camera
2. Horn BKP (1986) Robot vision. MIT, Cambridge
3. Xu G, Zhang Z (1996) Epipolar geometry in stereo, motion and object recognition. Kluwer Academic, Dordrecht
4. Pahlavan K, Uhlin T, Ekhlund JO (1993) Dynamic fixation. In: Proceedings of the 4th international conference on computer vision, Berlin, Germany. IEEE Computer Society, Los Alamitos, pp 412–419
5. Lavest J (1992) Stéréovision axiale par zoom pour la robotique. Ph.D. thesis, Université Blaise Pascal de Clermont-Ferrand, France



Camera Parameters (Internal, External)

► Camera Parameters (Intrinsic, Extrinsic)

Camera Parameters (Intrinsic, Extrinsic)

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Synonyms

Camera model; Camera parameters (internal, external)

Related Concepts

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

Definition

Camera parameters are the parameters used in a camera model to describe the mathematical relationship between the 3D coordinates of a point in the scene from which the light comes from and the 2D coordinates of its projection onto the image plane. The *intrinsic parameters*, also known as *internal parameters*, are the parameters intrinsic to the camera itself, such as the focal length and lens distortion. The *extrinsic parameters*, also known as *external parameters* or *camera pose*, are the parameters used to describe the transformation between the camera and its external world.

Background

In computer vision, in order to understand the environment surrounding us with a camera, we have to know first the camera parameters. Depending on the accuracy we need to achieve and on the quality of the

camera, some parameters can be neglected. For example, with a high-quality camera, the lens distortion can usually be ignored in most of the applications.

Theory

In the entry ► [Perspective Camera](#), we describe the mathematical model of a perspective camera with only a single parameter, the focal length f . The relationship between a 3D point and its image projection is described by

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}}, \quad (1)$$

where $s = S$ is an arbitrary nonzero scalar and \mathbf{P} is a *projective projection matrix* given by

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

Before proceeding, the reader needs to review the entry ► [Perspective Camera](#).

Extrinsic Parameters

In the projective projection matrix described in the entry ► [Perspective Camera](#), recapitulated above, we assumed that 3D points are expressed in the camera coordinate system. In practice, they can be expressed in any 3D coordinate system, which is sometimes referred as the *world coordinate system*. As shown in Fig. 1, we go from the old coordinate system centered at the optical center C (camera coordinate system) to the new coordinate system centered at point O (world coordinate system) by a rotation \mathbf{R} followed by a translation $\mathbf{t} = CO$. Then, for a single point, its coordinates expressed in the camera coordinate system, M_c , and those expressed in the world coordinate system, M_w , are related by

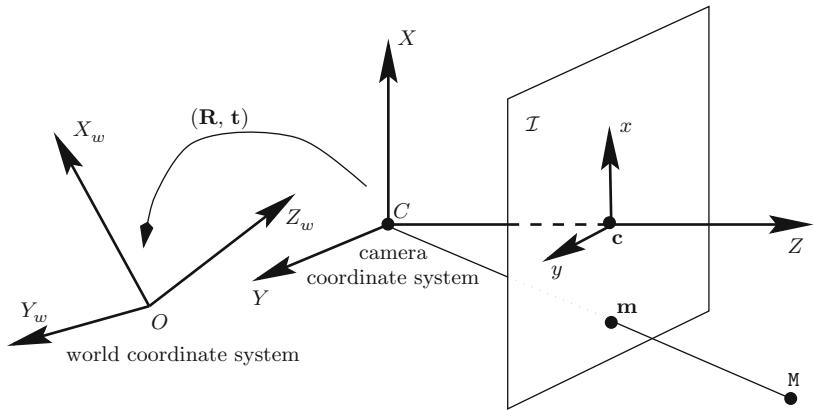
$$\mathbf{M}_c = \mathbf{RM}_w + \mathbf{t},$$

or more compactly

$$\tilde{\mathbf{M}}_c = \mathbf{D}\tilde{\mathbf{M}}_w, \quad (2)$$

Camera Parameters**(Intrinsic, Extrinsic), Fig. 1**

World coordinate system and camera extrinsic parameters



where \mathbf{D} is a Euclidean transformation of the three-dimensional space

$$\mathbf{D} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_3^T & 1 \end{bmatrix} \quad \text{with } \mathbf{0}_3 = [0, 0, 0]^T. \quad (3)$$

The matrix \mathbf{R} and the vector \mathbf{t} describe the orientation and position of the camera with respect to the new world coordinate system. They are called the *extrinsic parameters* of the camera.

From (1) and (2), we have

$$\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}}_c = \mathbf{P}\mathbf{D}\tilde{\mathbf{M}}_w.$$

Therefore the new perspective projection matrix is given by

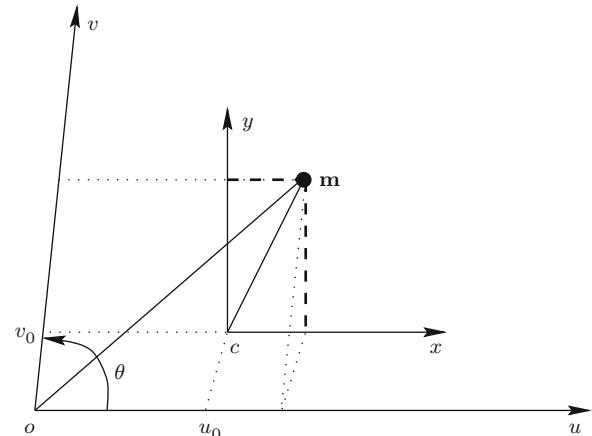
$$\mathbf{P}_{\text{new}} = \mathbf{P}\mathbf{D}. \quad (4)$$

This tells us how the perspective projection matrix \mathbf{P} changes when we change coordinate systems in the three-dimensional space: We simply multiply it on the right by the corresponding Euclidean transformation.

Intrinsic Parameters and Normalized Camera

This section considers the transformation in image coordinate systems. It is very important in practical applications because:

- We do not know the origin of the image plane in advance. It generally does not coincide with the intersection of the optical axis and the image plane.
- The units of the image coordinate axes are not necessarily equal, and they are determined by the sampling rates of the imaging devices.



Camera Parameters (Intrinsic, Extrinsic), Fig. 2 Camera intrinsic parameters

- The two axes of a real image may not form a right angle.

To handle these effects, we introduce an affine transformation.

Consider Fig. 2. The original image coordinate system \$(c, x, y)\$ is centered at the principal point \$c\$ and has the same units on both \$x\$- and \$y\$-axes. The coordinate system \$(o, u, v)\$ is the coordinate system in which we address the pixels in an image. It is usually centered at the upper left corner of the image, which is usually not the principal point \$c\$. Due to the electronics of acquisition, the pixels are usually not square. Without loss of generality, the \$u\$-axis is assumed to be parallel to the \$x\$-axis. The units along the \$u\$- and \$v\$-axes are assumed to be \$k_u\$ and \$k_v\$ with respect to the unit used in \$(c, x, y)\$. The \$u\$- and \$v\$-axes may not be exactly orthogonal, and we denote their angle by \$\theta\$. Let the coordinates of the

principal point c in (o, u, v) be $[u_0, v_0]^T$. These five parameters do not depend on the position and orientation of the cameras and are thus called the camera *intrinsic parameters*.

For a given point, let $\mathbf{m}_{\text{old}} = [x, y]^T$ be the coordinates in the original coordinate system; let $\mathbf{m}_{\text{new}} = [u, v]^T$ be the pixel coordinates in the new coordinate system. It is easy to see that

$$\tilde{\mathbf{m}}_{\text{new}} = \mathbf{H} \tilde{\mathbf{m}}_{\text{old}},$$

where

$$\mathbf{H} = \begin{bmatrix} k_u & k_u \cot \theta & u_0 \\ 0 & k_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Since, according to (1), we have

$$s \tilde{\mathbf{m}}_{\text{old}} = \mathbf{P}_{\text{old}} \tilde{\mathbf{M}},$$

we conclude that

$$s \tilde{\mathbf{m}}_{\text{new}} = \mathbf{H} \mathbf{P}_{\text{old}} \tilde{\mathbf{M}},$$

and thus

$$\mathbf{P}_{\text{new}} = \mathbf{H} \mathbf{P}_{\text{old}} = \begin{bmatrix} f k_u & f k_u \cot \theta & u_0 & 0 \\ 0 & f k_v / \sin \theta & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (5)$$

Note that it depends on the products $f k_u$ and $f k_v$, which means that a change in the focal length and a change in the pixel units are indistinguishable. We thus introduce two parameters $\alpha_u = f k_u$ and $\alpha_v = f k_v$.

We will now define a special coordinate system that allows us to normalize the image coordinates [1]. This coordinate system is called the *normalized coordinate system* of the camera. In this “normalized” camera, the image plane is located at a unit distance from the optical center (*i.e.* $f = 1$). The perspective projection matrix of the normalized camera is given by

$$\mathbf{P}_N = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (6)$$

For a world point $[X, Y, Z]^T$ expressed in the camera coordinate system, its normalized coordinates are

$$\begin{aligned} \hat{x} &= \frac{X}{Z} \\ \hat{y} &= \frac{Y}{Z}. \end{aligned} \quad (7)$$

A matrix \mathbf{P} defined by (5) can be decomposed into the product of two matrices:

$$\mathbf{P}_{\text{new}} = \mathbf{A} \mathbf{P}_N, \quad (8)$$

where

$$\mathbf{A} = \begin{bmatrix} \alpha_u & \alpha_u \cot \theta & u_0 \\ 0 & \alpha_v / \sin \theta & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (9)$$

The matrix \mathbf{A} contains only the intrinsic parameters and is called *camera intrinsic matrix*. It is thus clear that the normalized image coordinates are given by

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \end{bmatrix} = \mathbf{A}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \quad (10)$$

Through this transformation from the available pixel image coordinates, $[u, v]^T$, to the imaginary normalized image coordinates, $[\hat{x}, \hat{y}]^T$, the projection from the space onto the normalized image does not depend on the specific cameras. This frees us from thinking about characteristics of the specific cameras and allows us to think in terms of ideal systems in stereo, motion, and object recognitions.

The General Form of Perspective Projection Matrix

The camera can be considered as a system that depends upon the intrinsic and the extrinsic parameters. There are five intrinsic parameters: the scale factors α_u and α_v , the coordinates u_0 and v_0 of the principal point, and the angle θ between the two image axes. There are six extrinsic parameters, three for the rotation and three for the translation, which define the transformation from the world coordinate system to the standard coordinate system of the camera.

Combining (4) and (8) yields the general form of the perspective projection matrix of the camera:

$$\mathbf{P} = \mathbf{A} \mathbf{P}_N \mathbf{D} = \mathbf{A} [\mathbf{R} \ \mathbf{t}]. \quad (11)$$

The projection of 3D world coordinates $\mathbf{M} = [X, Y, Z]^T$ to 2D pixel coordinates $\mathbf{m} = [u, v]^T$ is then described by

$$s\tilde{\mathbf{m}} = \mathbf{P}\tilde{\mathbf{M}}, \quad (12)$$

where s is an arbitrary scale factor. Matrix \mathbf{P} has $3 \times 4 = 12$ elements but has only 11 degrees of freedom because it is defined up to a scale factor.

Let p_{ij} be the (i, j) entry of matrix \mathbf{P} . Eliminating the scalar s in (12) yields two nonlinear equations:

$$u = \frac{p_{11}X + p_{12}Y + p_{13}Z + p_{14}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}} \quad (13)$$

$$v = \frac{p_{21}X + p_{22}Y + p_{23}Z + p_{24}}{p_{31}X + p_{32}Y + p_{33}Z + p_{34}}. \quad (14)$$

Camera *calibration* is the process of estimating the intrinsic and extrinsic parameters of a camera or the process of first estimating the matrix \mathbf{P} and then deducing the camera parameters from \mathbf{P} . A wealth of work has been carried out in this domain by researchers either in photogrammetry [2, 3] or in computer vision and robotics [4–9] (see [10] for a review). The usual method of calibration is to compute camera parameters from one or more images of an object of *known size and shape*, for example, a flat plate with a regular pattern marked on it. From (12) or (13) and (14), we have two nonlinear equations relating 2D to 3D coordinates. This implies that each pair of an identified image point and its corresponding point on the calibration object provides two constraints on the intrinsic and extrinsic parameters of the camera. The number of unknowns is 11. It can be shown that, given N points ($N \geq 6$) in general position, the camera can be calibrated. The presentation of calibration techniques is beyond the scope of this book. The interested reader is referred to the above-mentioned references.

Once the perspective projection matrix \mathbf{P} is given, we can compute the coordinates of the optical center C of the camera in the world coordinate system. We first decompose the 3×4 matrix \mathbf{P} as the concatenation of a 3×3 submatrix \mathbf{B} and a 3-vector \mathbf{b} , that is, $\mathbf{P} = [\mathbf{B} \ \mathbf{b}]$. Assume that the rank of \mathbf{B} is 3. In the entry ▶ **Perspective Camera**, we explained that, under the pinhole model, the optical center projects to $[0, 0, 0]^T$ (*i.e.* $s = 0$). Therefore, the optical center can

be obtained by solving

$$\mathbf{P}\tilde{\mathbf{C}} = \mathbf{0}, \quad \text{that is, } [\mathbf{B} \ \mathbf{b}] \begin{bmatrix} C \\ 1 \end{bmatrix} = \mathbf{0}.$$

The solution is

$$\mathbf{C} = -\mathbf{B}^{-1}\mathbf{b}. \quad (15)$$

Given matrix \mathbf{P} and an image point \mathbf{m} , we can obtain the equation of the 3-D semi-line defined by the optical center C and point \mathbf{m} . This line is called the *optical ray* defined by \mathbf{m} . Any point on it projects to the single point \mathbf{m} . We already know that C is on the optical ray. To define it, we need another point. Without loss of generality, we can choose the point D such that the scale factor $s = 1$, that is,

$$\tilde{\mathbf{m}} = [\mathbf{B} \ \mathbf{b}] \begin{bmatrix} D \\ 1 \end{bmatrix}.$$

This gives $D = \mathbf{B}^{-1}(-\mathbf{b} + \tilde{\mathbf{m}})$. A point on the optical ray is thus given by

$$\mathbf{M} = C + \lambda(D - C) = \mathbf{B}^{-1}(-\mathbf{b} + \lambda\tilde{\mathbf{m}}),$$

where λ varies from 0 to ∞ .

References

1. Faugeras O (1993) Three-dimensional computer vision: a geometric viewpoint. MIT, Cambridge
2. Brown DC (1971) Close-range camera calibration. Photogramm Eng 37(8):855–866
3. Faig W (1975) Calibration of close-range photogrammetry systems: mathematical formulation. Photogramm Eng Remote Sens 41(12):1479–1486
4. Tsai R (1986) Multiframe image point matching and 3d surface reconstruction. IEEE Trans Pattern Anal Mach Intell 5:159–174
5. Faugeras O, Toscani G (1986) The calibration problem for stereo. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Miami Beach, FL. IEEE, Los Alamitos, pp 15–20
6. Lenz R, Tsai R (1987) Techniques for calibrating of the scale factor and image center for high accuracy 3D machine vision metrology. In: International conference on robotics and automation, Raleigh, NC, pp 68–75
7. Toscani G (1987) Système de Calibration optique et perception du mouvement en vision artificielle. Ph.D. thesis, Paris-Orsay
8. Wei G, Ma S (1991) Two plane camera calibration: a unified model. In: Proceedings of the IEEE conference on



- computer vision and pattern recognition (CVPR), Hawaii, pp 133–138
9. Weng J, Cohen P, Rebibo N (1992) Motion and structure estimation from stereo image sequences. IEEE Trans RA 8(3):362–382
 10. Tsai R (1989) Synopsis of recent progress on camera calibration for 3D machine vision. In: Khatib O, Craig JJ, Lozano-Pérez T (eds) The robotics review. MIT, Cambridge, pp 147–159

Camera Pose

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Synonyms

Camera extrinsic parameters

Related Concepts

- Camera Calibration; ► Camera Parameters (Intrinsic, Extrinsic); ► Intrinsic; ► Perspective Camera

Definition

Camera pose is referred to the position and orientation of a camera with respect to a reference coordinate system, which is usually known as the world coordinate system.

Background

Determining the camera pose is usually a first step toward perceiving the surrounding environment. In structure from motion where a camera is moving through the environment, one needs to determine the successive camera poses at different instants in order to reconstruct the surrounding environment in 3D. In a multi-camera (two or more) system, one needs to determine the relative camera pose, i.e., how one camera is related to other cameras.

The reader is referred to entry “► Camera Parameters (Intrinsic, Extrinsic)” for details.

Camera Response Function

- Radiometric Response Function

Camera-Shake Blur

- Motion Blur

Catadioptric Camera

Srikumar Ramalingam
Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

Synonyms

- Catoptrics; ► Dioptrics

Related Concepts

- Center of Projection; ► Field of View;
- Omnidirectional Camera

Definition

A catadioptric system is a camera configuration where both lenses and mirrors are jointly used to achieve specialized optical properties. These configurations are referred to as *catadioptric*, where “cata” comes from mirrors (reflective) and “dioptric” comes from lenses (refractive).

Background

In 1637, René Descartes observed that the refractive and reflective “ovals” (conical lenses and mirrors) have the ability to focus light into one single point on illumination from a chosen point [1]. It was reported that the same results were derived by Feynman et al. [2] and



Drucker and Locke [3]. In computer vision community, Baker and Nayar presented the complete class of single viewpoint catadioptric configurations with detailed solutions and degenerate cases [4]. Some of these results have been independently derived by Bruckstein and Richardson [5]. Survey of various catadioptric cameras, a review and details of their calibration, and 3D reconstruction algorithms can also be found in [6, 7].

Theory and Classification

The combination of mirrors and lenses provides a wide range of design possibilities leading to interesting applications in computer vision. Most catadioptric configurations have larger field of view compared to conventional pinhole cameras. Other important design goals are compactness of a sensor, a single effective viewpoint, image quality, focusing properties, or a desired projection function. The catadioptric cameras may be classified in many ways. In [6], Sturm et al. classify the catadioptric cameras in to five different types:

- Single-mirror central systems, having a single effective viewpoint
- Central systems using multiple mirrors
- Noncentral systems
- Single-lens stereo systems
- Programmable devices

In what follows, catadioptric cameras are classified into central and noncentral systems. Most of these catadioptric configurations were proposed by researchers along with specified calibration and 3D reconstruction algorithms.

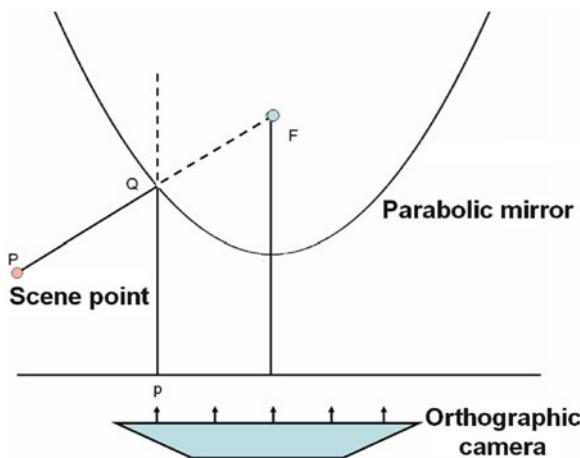
Central Catadioptric Configurations

It requires a very careful choice of the shape of the mirrors and their positioning to obtain a single effective viewpoint in catadioptric imaging. The single viewpoint design goal is important because it allows the generation of pure perspective images from the catadioptric images. Furthermore, it allows one to solve motion estimation and 3D reconstruction algorithms in the same way as perspective cameras:

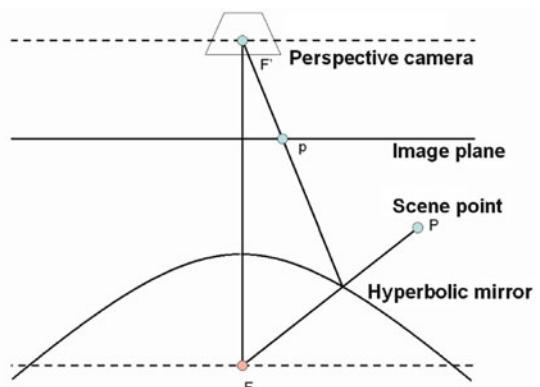
- *Planar mirror*: In [4, 8], it can be observed that by using planar mirrors along with a perspective camera, one can obtain a single viewpoint configuration.

Since planar mirrors do not increase the field of view of the system, they are not very interesting for building omnidirectional cameras. Using four planar mirrors in a pyramidal configuration along with four perspective cameras, Nalwa [9] produced an omnidirectional sensor of field of view of $360^\circ \times 50^\circ$. The optical centers of the four cameras and the angles made by the four planar faces are adjusted to obtain a single effective viewpoint for the system.

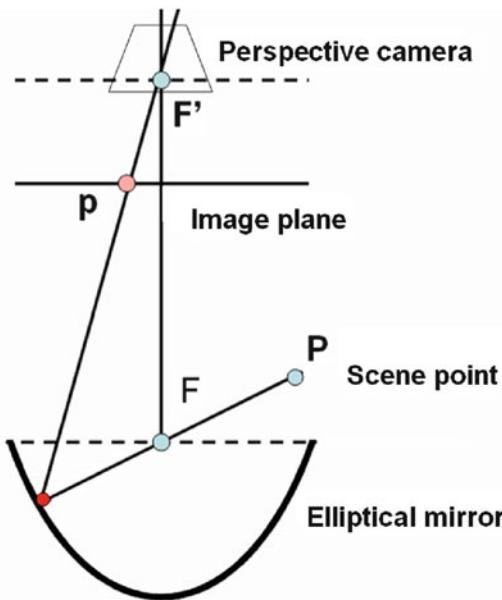
- *Conical mirrors*: By positioning the optical center of a perspective camera at the apex of a cone, one can obtain a single center configuration. Nevertheless, the only light rays reaching the camera after a reflection in the mirror are those grazing the cone. This case is thus not useful to enhance the field of view while conserving a single center of projection. However, in the work [10], it was proved that conical mirrors can be used to construct a non-degenerate single viewpoint omnidirectional cameras. The outer surface of the conical mirror forms a virtual image corresponding to the real scene behind the conical mirror. On placing the optical center of the pinhole camera at the vertex of the cone, the camera sees the world through the reflection on the outer surface of the mirror. In other words, the cone is not blocking the view. On the other hand, the cone is the view.
- *Spherical mirror*: If the optical center of a perspective camera is fixed at the center of a spherical mirror, one can obtain a single viewpoint configuration. Unfortunately, all that the perspective camera sees is its own reflection. As a result the spherical mirror produces a degenerate configuration without any advantage. Remember that by positioning the perspective camera outside the sphere, one can obtain a useful *noncentral* catadioptric camera.
- *Parabolic mirror*: Figure 1 shows a single viewpoint catadioptric system with a parabolic mirror and an orthographic camera. It is easier to study a catadioptric configuration by considering the back projection rather than the forward projection. Consider the back projection of an image point \mathbf{p} . The back-projected ray from the image pixel \mathbf{p} , starting from the optical center at infinity, is parallel to the axis of the parabolic mirror. This ray intersects and reflects from the surface of the mirror. The reflection is in accordance with the laws of reflection. This



Catadioptric Camera, Fig. 1 Parabolic mirror + orthographic camera [8]. **P** refers to the 3D scene point. **F**, the focus of the parabolic mirror, is the effective viewpoint



Catadioptric Camera, Fig. 3 Hyperbolic mirror + perspective camera [4]. **P** refers to the 3D scene point. **F** and **F'** refer to the two foci of the mirror and **p** refers to the image point. **F** is the effective viewpoint



Catadioptric Camera, Fig. 2 Elliptical mirror + perspective camera [4]. **P** refers to the 3D scene point. **F** and **F'** refer to the two foci of the mirror and **p** refers to the image point. **F** is the effective viewpoint

reflected light ray is nothing but the incoming light ray from a scene point **P** in forward projection. The incoming ray passes through the focus **F** if extended on the inside of the mirror. This point where all the incoming light rays intersect (virtually) is called the effective viewpoint.

– *Elliptical mirror:* Figure 2 shows a central catadioptric system with an elliptical mirror and a perspective camera. The optical center of the perspective camera is placed at the upper focus of the elliptical mirror. By back-projecting an image point **p**, one can observe the following. The back-projected ray, starting from the optical center at the upper focus of the elliptical mirror, intersects and reflects from the surface of the elliptical mirror. The reflected back-projected ray, or the incoming light ray, virtually passes through the lower focus of the mirror. Thus, the lower focus (**F**) is the effective viewpoint of the system.

– *Hyperbolic mirror:* In Fig. 3, a catadioptric system is shown with a hyperbolic mirror and a perspective camera. The optical center of the perspective camera is placed at the external focus of the mirror **F'**. The back-projected ray of the image point **p** starts from the optical center, which is the external focus **F'** of the mirror, of the perspective camera. Using the same argument as above, one can observe that the lower focus **F** is the effective viewpoint. The first known work to use a hyperbolic mirror along with a perspective camera at the external focus of the mirror to obtain a single effective viewpoint configuration is [11]. Later in 1995, a similar implementation was proposed in [12].

Noncentral Catadioptric Cameras

Single viewpoint configurations are extremely delicate to construct, handle, and maintain. By relaxing

this single viewpoint constraint, one can obtain greater flexibility in designing novel systems. In fact most real catadioptric cameras are geometrically noncentral, and even the few restricted central catadioptric configurations are usually noncentral in practice [13]. For example, in the case of para-catadioptric cameras, the telecentric lens is never truly orthographic and it is difficult to precisely align the mirror axis and the axis of the camera. In hyperbolic or elliptic configurations, precise positioning of the optical center of the perspective camera in one of the focal points of the hyperbolic or elliptic mirror is practically infeasible. In [14], Ramalingam et al. show that most of the practically used catadioptric configurations fall under an axial camera model where all the projection rays pass through a single line rather than a single point in space. A few noncentral catadioptric configurations are mentioned below. Analogous to the single viewpoint in central cameras, there is a *viewpoint locus* in non-central cameras. It can be defined as follows: a curve or other set of points such that all projection rays cut at least one of the points in the viewpoint locus. Usually, one tries to find the “simplest” such set of points:

- *Conical mirror*: On using a conical mirror in front of a perspective camera, one can obtain an omnidirectional sensor [15, 16]. Nevertheless this configuration does not obey the single viewpoint restriction (besides in the degenerate case of the perspective optical center being located at the cone’s vertex). If the optical center lies on the mirror axis, then the viewpoint locus is a circle in 3D, centered in the mirror axis (it can be pictured as a halo over the mirror). An alternative choice of viewpoint locus is the mirror axis. Otherwise, the viewpoint locus is more general.
- *Spherical mirror*: On using a spherical mirror along with a perspective camera, one can enhance the field of view of the imaging system [16–18]. Again this configuration does not obey the single viewpoint restriction (besides in the degenerate case of the perspective optical center being located at the sphere center).
- *Digital micro-mirror array*: Another interesting camera is the recently introduced programmable imaging device using a digital micro-mirror array [19]. A perspective camera is made to observe a scene through a programmable array of micro-mirrors. By controlling the orientations and

positions of these mirrors, one can obtain an imaging system with complete control (both in terms of geometric and radiometric properties) over the incoming light ray for every pixel. However, there are several practical issues which make it difficult to realize the full potential of such an imaging system. First, current hardware constraints prohibit the usage of more than two possible orientations for each micro-mirror. Second, arbitrary orientations of the micro-mirrors would produce a discontinuous image which is unusable for many image processing operations.

- *Oblique cameras*: An ideal example for a noncentral camera is an oblique camera. No two rays intersect in an oblique camera [20]. In addition to developing multi-view geometry for oblique cameras, Pajdla also proposed a physically realizable system which obeys oblique geometry. The practical system consists of a rotating catadioptric camera that uses a conical mirror and a telecentric optics. The viewpoint locus is equivalent to a two-dimensional surface or a set of points, where each of the projection rays passes through at least one of the points. Different catadioptric configurations come with different calibration and 3D reconstruction algorithms. Recently, there has been a lot of interest in unifying different camera models and developing generic calibration and 3D reconstruction algorithms [21–24].

Application

Due to enhanced field of view, catadioptric cameras are mainly used in surveillance, car navigation, image-based localization, and augmented reality applications.

References

1. Descartes R, Smith D (1637) The geometry of René Descartes. Dover, New York. Originally published in Discours de la Méthode
2. Feynman R, Leighton R, Sands M (1963) The feynman lectures on physics. Mainly mechanics, radiation, and heat, Addison-Wesley, Reading, vol 1
3. Drucker D, Locke P (1996) A natural classification of curves and surfaces with reflection properties. Math Mag 69(4):249–256
4. Baker S, Nayar S (1998) A theory of catadioptric image formation. In: International conference on computer vision (ICCV), Bombay, pp 35–42

5. Bruckstein A, Richardson T (2000) Omnidirectional cameras with curved surface mirrors. In: IEEE workshop on omnidirectional vision, Hilton Head Island, pp 79–84
6. 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
7. Ramalingam S (2006) Generic imaging models: calibration and 3d reconstruction algorithms. PhD thesis, INRIA Rhone Alpes
8. Nayar S (1997) Catadioptric omnidirectional camera. In: Conference on computer vision and pattern recognition (CVPR), San Juan, pp 482–488
9. Nalwa V (1996) A true omnidirectional viewer. Technical report, Bell Laboratories, Holmdel, NJ, USA
10. Lin S, Bajcky R (2001) True single view cone mirror omnidirectional catadioptric system. In: International conference on computer vision (ICCV), Vancouver, vol 2, pp 102–107
11. Rees DW (1970) Panoramic television viewing system. US Patent 3,505,465
12. Yamazawa K, Yagi Y, Yachida M (1995) Obstacle avoidance with omnidirectional image sensor hyperomni vision. In: International conference on robotics and automation, Nagoya, pp 1062–1067
13. Micusik B, Pajdla T (2004) Autocalibration and 3d reconstruction with non-central catadioptric cameras. In: Computer vision and pattern recognition (CVPR), San Diego, pp 748–753
14. Ramalingam S, Sturm P, Lodha S (2006) Theory and calibration algorithms for axial cameras. In: ACCV, Hyderabad
15. Yagi Y, Kawato S (1990) Panoramic scene analysis with conic projection. In: International conference on robots and systems (IROS), Cincinnati
16. Bogner S (1995) Introduction to panoramic imaging. In: IEEE SMC, New York, vol 54, pp 3100–3106
17. Hong J (1991) Image based homing. In: International conference on robotics and automation, Sacramento
18. Murphy JR (1995) Application of panoramic imaging to a teleoperated lunar rover. In: IEEE SMC conference, Vancouver, pp 3117–3121
19. Nayar S, Branzoi V, Boult T (2004) Programmable imaging using a digital micromirror array. In: International conference on computer vision and pattern recognition (CVPR), Washington, DC, pp 436–443
20. Pajdla T (2002) Stereo with oblique cameras. Int J Comput Vis 47(1):161–170
21. Geyer C, Daniilidis K (2000) A unifying theory of central panoramic systems and practical implications. In: European conference on computer vision (ECCV), Dublin, pp 159–179
22. Grossberg M, Nayar S (2001) A general imaging model and a method for finding its parameters. In: International conference on computer vision (ICCV), Vancouver, vol 2, pp 108–115
23. Sturm P, Ramalingam S (2004) A generic concept for camera calibration. In: European conference on computer vision (ECCV), Prague, vol 2, pp 1–13
24. Ramalingam S, Sturm P, Lodha S (2005) Towards complete generic camera calibration. In: Conference on computer vision and pattern recognition (CVPR), San Diego

Catoptrics

► [Catadioptric Camera](#)

Center of Projection

Srikumar Ramalingam
Mitsubishi Electric Research Laboratories,
Cambridge, MA, USA

Synonyms

[Optical center; Single viewpoint](#)

Related Concepts

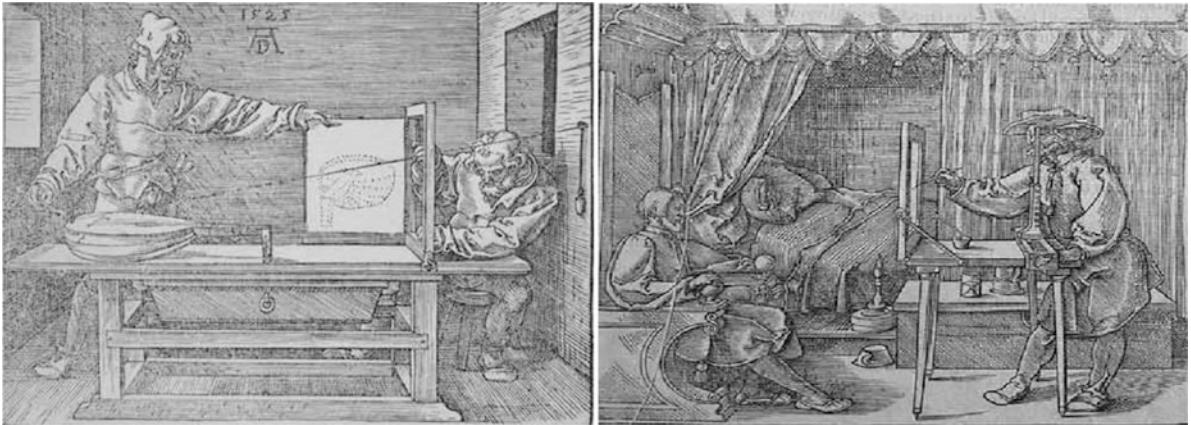
► [Field of View](#)

Definition

Center of projection is a single 3D point in space where all the light rays sampled by a conventional pinhole camera intersect.

Background

Albrecht Dürer, a German artist, published a treatise on measurement using a series of illustrations of drawing frames and perspective machines. On the left side of Fig. 1, an apparatus for drawing a lute is shown. One end of a thread is attached to a pointer and the other end to a pulley on a wall. The thread also passes through a frame in between the lute and the pulley. When the pointer is fixed at different points on the lute, the vertical and horizontal coordinates of the thread, as it passes through the frame, are marked. By meticulously marking the coordinates for each point on the lute, the perspective image of the lute is created. It is obvious to see the intuition behind this setup, i.e., its similarity to a pinhole camera. The pulley is equivalent to the single viewpoint or the center of projection, the frame replaces the image plane, and finally, the thread is nothing but the light ray emerging from the scene.



Center of Projection, Fig. 1 *Left:* One of Albrecht Dürer's perspective machines, which was used to draw a lute in the year 1525 [1]. *Right:* An artist uses the perspective principle

Though the principle is correct, the procedure is quite complicated. On the right side of Fig. 1, another perspective machine is shown. One can observe an artist squinting through a peep hole with one eye to keep a single viewpoint and tracing his sitter's features onto a glass panel. The idea is to trace the important features first and then transfer the drawing for further painting.

Theory

Pinhole Camera Model

Consider the perspective model that is shown in Fig. 2. Every 3D scene point $\mathbf{P}(X, Y, Z)$ gets projected onto the image plane to a point $\mathbf{p}(x, y)$ through the optical center \mathbf{C} . The optical axis is the perpendicular line to the image plane passing through the optical center. The center of radial symmetry in the image or principal point, i.e., the point of intersection of the optical axis and the image plane, is given by \mathbf{O} . The distance between \mathbf{C} (optical center) and the image plane is the focal length f . The optical center of the camera is the origin of the coordinate system. The image plane is parallel to the XY plane, held at a distance of f from the origin. Using the basic laws of trigonometry, one can observe the following:

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z}$$

to accurately draw a man sitting on a chair, by looking at him through a peep hole with one eye, and tracing his features on a glass plate

Once expressed in homogeneous coordinates, the above relations transform to the following:

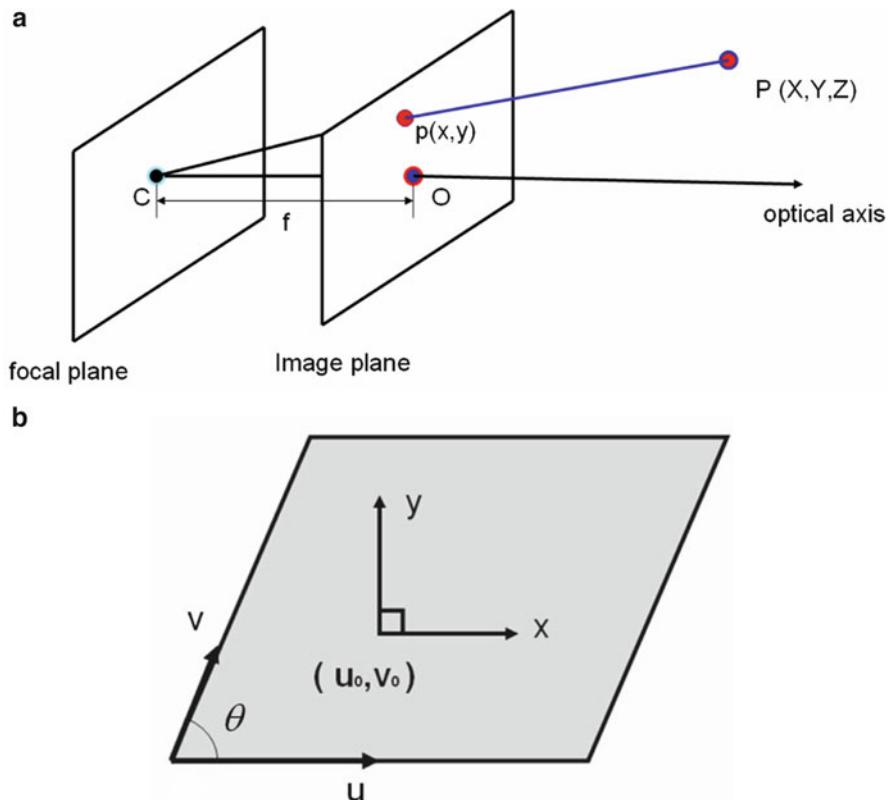
$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \sim \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

where the relationship \sim stands for “equal up to a scale.”

Practically available CCD cameras deviate from the perspective model. First, the principal point (u_0, v_0) does not necessarily lie on the geometrical center of the image. Second, the horizontal and vertical axes (u and v) of the image are not always perfect perpendicular. Let the angle between the two axes be θ . Finally, each pixel is not a perfect square and consequently, f_u and f_v are the two focal lengths that are measured in terms of the unit lengths along u and v directions. By incorporating these deviations in the camera model, one can obtain the following scene (X, Y, Z) to image (u, v) transformation:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_x & f_v \cot \theta & u_0 & 0 \\ 0 & \frac{f_v}{\sin \theta} & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

In practice, the 3D point is available in some world coordinate system that is different from the camera



Center of Projection, Fig. 2 (a) Perspective camera model. (b) The relationship between (u, v) and (x, y) is shown

coordinate system. The motion between these coordinate systems is given by (R, t) :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} f_x & f_v \cot \theta & u_0 \\ 0 & \frac{f_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} [R \quad -Rt] \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

$$M = \begin{bmatrix} f_x & f_v \cot \theta & u_0 \\ 0 & \frac{f_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix} [R \quad -Rt]$$

$$K = \begin{bmatrix} f_x & f_v \cot \theta & u_0 \\ 0 & \frac{f_v}{\sin \theta} & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

The 3×4 matrix M that projects a 3D scene point P to the corresponding image point p is called the projection matrix. The 3×3 matrix K that contains the internal parameters $(u_0, v_0, \theta, f_x, f_y)$ is generally referred to as the *intrinsic* matrix of a camera.

In back projection, given an image point p , the goal is to find the set of 3D points that project to it. The back projection of an image point is a ray in space. One can compute this ray by identifying two points on this ray. The first point can be the optical center C , since it lies on this ray. Since $MC = 0$, C is nothing but the right nullspace of M . Second, the point $M^+ p$, where M^+ is the pseudoinverse of M , lies on the back-projected ray because it projects to point p on the image. Thus, the back projection of p can be computed as follows:

$$P(\lambda) = M^+ p + \lambda C$$

The parameter λ allows to get different points on the back-projected ray.

Caustics

In a single viewpoint imaging system, the geometry of the projection rays is given by the effective viewpoint and the direction of the projection rays. In a noncentral imaging system, *caustic*, a well-known terminology in

the optics community, can be utilized for representing the geometry of projection rays [2]. A caustic refers to the loci of viewpoints in 3D space to represent a noncentral imaging system. Concretely, the envelope of all incoming light rays that are eventually imaged is defined as the caustic. A caustic is referred to as dia-caustic for dioptric (lens-based systems) and catacaustic (mirror-based systems) for catadioptric systems. A complete study of conic catadioptric systems has been done [3]. Once the caustic is determined, each point on the caustic represents a light ray by providing its position and the direction. Position is given by the point on the caustic, and orientation is related to the concept of tangent. [Figure 3](#) shows the caustic for several noncentral imaging systems. For a single viewpoint imaging system, the caustic is a degenerate one being a single point. Simple methods exist for the computation of the caustic from the incoming light rays such as local conic approximations [4] and the so-called Jacobian method [5]. A few examples for caustics are shown in [Fig. 3](#).

Generalized and Multi-perspective Imaging Models

Many novel camera models have multiple centers of projection and they cannot be explained by a simple parametric pinhole model. In computer vision, there has been significant interest in generalizing the camera models to reuse the existing calibration and 3D reconstruction algorithms for novel cameras. In order to do this, first renounce on parametric models and adopt the following very general model: a camera acquires images consisting of pixels; each pixel captures light that travels along a ray in 3D. The camera is fully described by:

- The coordinates of these rays (given in some local coordinate system).
- The mapping between pixels and rays; this is basically a simple indexing.

The generic imaging model is shown in [Fig. 4](#). This allows to describe all above models and virtually any camera that captures light rays traveling along straight lines. The above imaging model has already been used, in more or less explicit form, in various works [3, 6–16], and is best described in [6]. There are conceptual links to other works: acquiring an image with a camera of the general model may be seen as sampling the plenoptic function [17], and a light field [18] or

lumigraph [19] may be interpreted as a single image, acquired by a camera of an appropriate design. More details of generic imaging model, their calibration, and 3D reconstruction algorithms can also be found in [20].

Taxonomy of Generic Imaging Models

Central Model: All the projection rays go through a single point, the optical center. Examples are mentioned below:

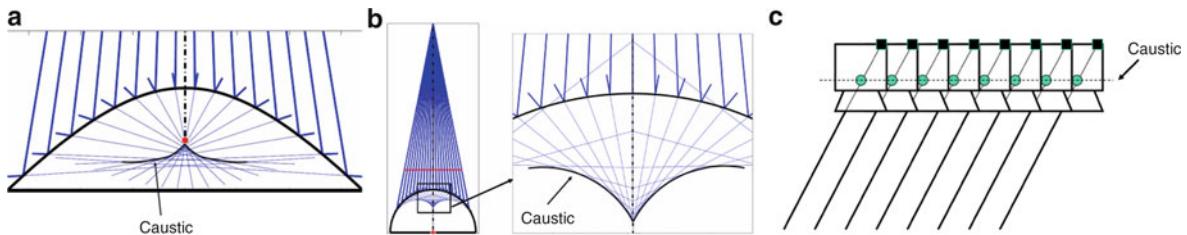
- The conventional perspective camera forms the classical example for a central camera.
- Perspective+radial or decentering distortion.
- Central catadioptric configurations using parabolic, hyperbolic, or elliptical mirrors.
- Fish-eye cameras can be considered as approximate central cameras.

Axial Model [21]: All the projection rays go through a single line in space, the *camera axis*. Examples of cameras falling into this class are:

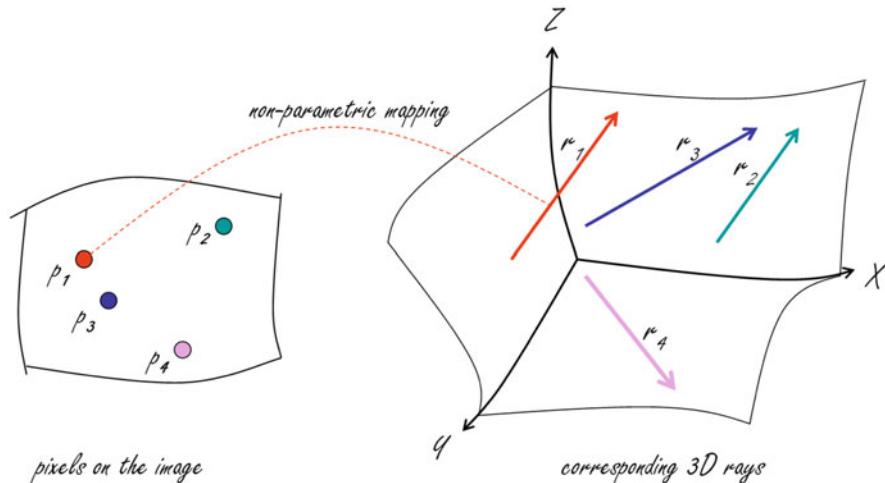
- Stereo systems consisting of 2, 3, or more central cameras with collinear optical centers.
- Noncentral catadioptric cameras of the following type: the mirror is any surface of revolution and the optical center of the central camera looking at it (can be any central camera, not only perspective) lies on its axis of revolution. It is easy to verify that in this case, all the projection rays cut the mirror's axis of revolution, i.e., the camera is an axial camera, with the mirror's axis of revolution as camera axis. Note that catadioptric cameras with a spherical mirror and a central camera looking at it are always axial ones.
- X-slit cameras [22] (also called two-slit or crossed-slit cameras), and their special case of linear push-broom cameras [23].

Noncentral Cameras: A noncentral camera may have completely arbitrary projection rays. Common examples are given below:

- Multi-camera system consisting of 3 or more cameras, all of whose optical centers are not collinear.
- Oblique camera: This is an ideal example for a non-central camera. No two rays intersect in an oblique camera [8].
- Imaging system using a micro-mirror array [24]. A perspective camera is made to observe a scene through a programmable array of micro-mirrors.



Center of Projection, Fig. 3 Caustics for several imaging systems (a) Hyperbolic catadioptric system (b) Spherical catadioptric system (c) Pushbroom camera



Center of Projection, Fig. 4 The main idea behind the generic imaging model: The relation between the image pixels (p_1, p_2, p_3, p_n) and their corresponding projection rays (r_1, r_2, r_3, r_n) is *non-parametric*

By controlling the orientations and positions of these mirrors, one can obtain an imaging system with complete control (both in terms of geometric and radiometric properties) over the incoming light ray for every pixel.

- Noncentral mosaic: An image sequence is captured by moving the optical center of a perspective camera in a circular fashion [3]. The center columns of the captured images are concatenated to create a noncentral mosaic image.
- Center strip mosaic: The optical center of the camera is moved [3]. The center columns of the captured images are concatenated to form a center strip mosaic. The resulting mosaic corresponds to a noncentral camera.

These three classes of camera models may also be defined as existence of a linear space of d dimensions that has an intersection with all the projection rays: $d = 0$ defines central, $d = 1$ axial, and $d = 2$ general noncentral cameras.

A detailed survey of various camera models, calibration, and 3D reconstruction algorithms is given in [25].

References

1. Dürer A (1525) Underweysung der Messung (Instruction in measurement), Book with more than 150 woodcuts
2. Born M, Wolf E (1965) Principles of optics. Permagon, Oxford
3. Swaminathan R, Grossberg M, Nayar S (2003) A perspective on distortions. In: Conference on computer vision and pattern recognition (CVPR), Madison, vol 2, p 594
4. Bruce JW, Giblin PJ, Gibson CG (1981) On caustics of plane curves. Am Math Mon 88:651–667
5. Burkhard D, Shealy D (1973) Flux density for ray propagation in geometrical optics. J Opt Soc Am 63(2):299–304
6. Grossberg M, Nayar S (2001) A general imaging model and a method for finding its parameters. In: International conference on computer vision (ICCV), Vancouver, vol 2, pp 108–115

7. Neumann J, Fermüller C, Aloimonos Y (2003) Polydioptropic camera design and 3d motion estimation. In: Conference on computer vision and pattern recognition (CVPR), Madison, vol 2, pp 294–301
8. Pajdla T (2002) Stereo with oblique cameras. Int J Comput Vis 47(1):161–170
9. Peleg S, Ben-Ezra M, Pritch Y (2001) Omnistereo: panoramic stereo imaging. IEEE Trans Pattern Anal Mach Intell 23:279–290
10. Pless R (2003) Using many cameras as one. In: Conference on computer vision and pattern recognition (CVPR), Madison, pp 587–594
11. Seitz S (2001) The space of all stereo images. In: International conference on computer vision (ICCV), Vancouver, vol 1, pp 26–33
12. Shum H, Kalai A, Seitz S (1999) Omnivergent stereo. In: International conference on computer vision (ICCV), Corfu, pp 22–29
13. Sturm P, Ramalingam S (2004) A generic concept for camera calibration. In: European conference on computer vision (ECCV), Prague, vol 2, pp 1–13
14. Ramalingam S, Sturm P, Lodha S (2005) Towards complete generic camera calibration. In: Conference on computer vision and pattern recognition (CVPR), San Diego
15. Wexler Y, Fitzgibbon A, Zisserman A (2003) Learning epipolar geometry from image sequences. In: Conference on computer vision and pattern recognition (CVPR), Madison, vol 2, pp 209–216
16. Wood D, Finkelstein A, Hughes J, Thayer C, Salesin D (1997) Multiperspective panoramas for cell animation. In: SIGGRAPH, Los Angeles, pp 243–250
17. Adelson E, Bergen J (1991) The plenoptic function and the elements of early vision. In: Computational models of visual processing, edited by Michael S. Landy and J. Anthony Movshon, MIT Press, Cambridge, MA, pp 3–20
18. Levoy M, Hanrahan P (1996) Light field rendering. In: SIGGRAPH, New Orleans, pp 31–42
19. Gortler S, Grzeszczuk R, Szeliski R, Cohen M (1996) The Lumigraph. In: SIGGRAPH, New Orleans, pp 43–54
20. Ramalingam S (2006) Generic imaging models: calibration and 3d reconstruction algorithms. PhD Thesis, INRIA Rhone Alpes
21. Ramalingam S, Sturm P, Lodha S (2006) Theory and calibration algorithms for axial cameras. In: ACCV, Hyderabad
22. Feldman D, Pajdla T, Weinshall D (2003) On the epipolar geometry of the crossed-slits projection. In: International conference on computer vision (ICCV), Nice
23. Gupta R, Hartley R (1997) Linear pushbroom cameras. In: IEEE transactions on pattern analysis and machine intelligence (PAMI) 19(9):963–975
24. Nayar S, Branzoi V, Boult T (2004) Programmable imaging using a digital micromirror array. In: International conference on computer vision and pattern recognition (CVPR), Washington, pp 436–443
25. Sturm P, Ramalingam S, Tardif JP, Gasparini S, Barreto J (2011) Camera models and fundamental concepts used in geometric computer vision. Found Trends Comp Gr Vis 6(1–2):1–183

Change Detection

Joseph L. Mundy

Division of Engineering, Brown University
Rensselaer Polytechnic Institute, Providence,
RI, USA

Synonyms

Normalcy modeling

Definition

A determination that there are significant differences between visual scenes

Background

Change detection is a key task for computer vision algorithms. The goal is to compare two or more visual scenes and report any significant differences between the scenes. As with many vision tasks, the meaning of *significant* is application dependent. The change detection task can be rendered somewhat more concrete by considering the types of changes that are not typically of interest. Examples of changes that are usually irrelevant are:

- Camera viewpoint
- Varying illumination
- Wind-based motion
- Weather, e.g., snow and rain

The implementation of algorithms that can detect interesting changes while ignoring trivial changes such as these is a very difficult problem, and only quite limited change detection capabilities have achieved to date. It is also the case that the change detection task, when viewed broadly, overlaps the scope of many other vision tasks such as visual inspection and moving object detection.

Early attempts at change detection were based on simple strategies such as thresholding the magnitude of intensity differences between a reference image and an image that manifests change. This simple approach is only practical if the scene and imaging conditions are



closely controlled so that the only scene changes are due to events of interest. Such highly controlled conditions can be found in industrial applications where the lighting and camera pose is accurately maintained. For example, missing components in a circuit board can be detected by comparing the image intensity of a high-quality *master* board with images of boards with potentially missing components or other flaws. In this application, images are accurately registered to the master image, so that intensity differences correspond to physical differences in the boards. The detected changes correspond to missing components or additional material such as excess solder.

A more advanced change detection principle is to classify elements of a scene into categories of interest. Then two scenes can be compared as to the presence or absence of various category instances. Change detection based on this principle can be successful if the classification process is insensitive to the types of irrelevant scene variations mentioned above. This classification approach has been applied extensively to the detection of changes in multispectral aerial and satellite imagery [1]. Multispectral images typically have four or more color bands, so that each pixel is a feature vector that can be subjected to standard classifiers. Image regions are classified into types such as roads, vegetation, forest, and water. Images can then be compared to determine the change in area of each category. This approach requires pixel-level image-to-image registration, which in turn is only effective if the viewpoint change between the images is small and the spatial resolution of the images is low.

Given the complexity of scene appearance and contextual variations, the great majority of change detection algorithms have adopted a learning approach based on the observation of a number of images. In this learning paradigm, it is assumed that changes are rare so that a statistical model for *normal* image appearance can be formed without removing regions of the image that represent change.

A good example of this principle is learning the normal appearance of a road surface in a set of images that depict moving vehicles on the road. In any given image, some of the road surface is visible and contributes correct image appearance information to the model. In other images different parts of the surface may be visible. The assumption is that over a large set of training images, the frequency of vehicle appearance

at any given road surface point will be small compared to the frequency of the road appearance. This approach can also be denoted as *background modeling* because a model is being constructed for the scene background rather than the moving or changing objects, i.e., *foreground*. The approach can also be considered as developing a model for *normalcy* where changes are infrequent, i.e., abnormal.

This statistical modeling principle has received widespread acceptance with the advent of cheap computing power and the significance of video data, which provides large training samples. A classical reference is the work of Stauffer and Grimson, who implemented a moving object detection system in online video streams [2]. An important aspect of change detection in video is that illumination and atmospheric properties vary relatively slowly compared to the video frame rate. Under this condition, statistical appearance models can gradually adapt to such variations and not manifest them as change.

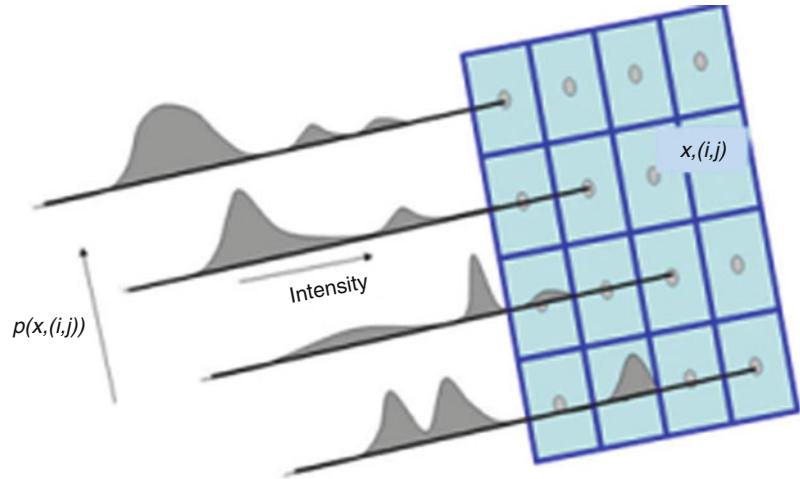
The effects of viewpoint and illumination can be overcome through the use of active 3-d range sensors such as LIDAR (Light Detection And Ranging) or laser triangulation. Scene illumination is provided by the sensor itself and therefore has known direction and spatial extent. The result is a 3-d point cloud of samples from scene surfaces and associated surface reflectance values. In this approach, change is detected by measuring the distance between two 3-d scene point sets after they have been accurately registered together. Missing points or points that are significantly far from the reference point set are considered to be change. Such change points can be grouped into a connected region to further characterize the change. An example of this strategy is the work of Girardeau-Montaut et al. [3].

Theory

The applications of change detection are so broad that a complete theoretical background is beyond the scope of an encyclopedia entry. Moreover, the success of change detection is critically dependent on accurate registration of the reference and current scenes so that change is due solely to the actual scene differences. However, the registration of scene images or 3-d data is a topic in its own right and will not be considered

Change Detection, Fig. 1

The appearance of each pixel is modeled by a Gaussian mixture distribution



here. Instead, two important statistical methods will be described that have achieved considerable success in image-based change detection: the joint probability method by Carlotto [4] and the background modeling method by Stauffer and Grimson [2].

Joint Probability Method

This approach can accommodate scenes with significant differences in illumination, and even between images taken with different sensor modalities such as visible and IR wave lengths. The method typically is applied to two images, $x_1(i, j)$ and $x_2(i, j)$, where the pixel intensity values, x , are considered to be random variables. It is assumed that the images are registered and the joint histogram, $p(x_1, x_2)$, of the intensity values at each pixel is accumulated. The expected intensity of a pixel in image 2, given the image value in image 1, is defined by

$$\tilde{x}_2(x_1) = \int x_2 p(x_2|x_1) dx_2, \quad (1)$$

where

$$p(x_2|x_1) = \frac{p(x_1, x_2)}{p(x_1)}.$$

Change is then defined as values of

$$c_{21}(i, j) = \|x_2(i, j) - \tilde{x}_2(x_1(i, j))\| \quad (2)$$

that exceed a decision threshold. The same analysis can be applied in reverse to compute

$$c_{12}(i, j) = \|x_1(i, j) - \tilde{x}_1(x_2(i, j))\|. \quad (3)$$

Gaussian Mixture Method

A second approach to modeling normal scene appearance is to associate a probability distribution with each image pixel location. This distribution accounts for normal variations in image intensity for each pixel across a set of images. It is assumed that the images are spatially registered so that a pixel in each image corresponds to the same surface element in the scene. An example of such registration is provided by a fixed video camera viewing a dynamic scene with moving objects and a stationary background.

The intensity of a given pixel will sometimes be due to background and sometimes foreground moving objects. Thus, an appropriate probability distribution for the overall appearance variation is a Gaussian mixture as shown in Fig. 1. The mixture distribution is defined by

$$p(x) = \sum_i \frac{w_i}{\sqrt{2\pi}\sigma_i} e^{-\frac{1}{2}\left(\frac{(x-\mu_i)}{\sigma_i}\right)^2}. \quad (4)$$

The distribution parameters, $\{w_i, \mu_i, \sigma_i\}$, are learned using a continuous online update algorithm where a new intensity sample, $x_{N+1}(i, j)$, is associated with an existing mixture component if it is within a few standard deviations of the component mean or a new



mixture is initiated. The update procedure for a scalar pixel intensity sample after N observations is defined by the following equations:

$$\begin{aligned} w_{N+1} &= \frac{Nw_N}{N+1} + \frac{1}{N+1} \\ \mu_{N+1} &= \frac{N\mu_N}{N+1} + \frac{x_N}{N+1} \\ \sigma_{N+1}^2 &= \frac{N\sigma_N^2}{N+1} + \frac{(x_N - \mu_N)^2}{N+1}. \end{aligned} \quad (5)$$

The update is applied if the sample, x_{N+1} , is within a specified number of standard deviations from one of the mixture component means. If a new sample is not within the capture range of one of the mixture components, then a new mixture component is added with user-specified default weight, w_i , and standard

deviation. When the limit on the number of mixture components is reached, the component with the smallest value of $\frac{w_i}{\sigma_i}$ is discarded. This component is the least informative (greatest entropy) of the mixture.

The update can be applied continuously over a video sequence, and the mixture parameters will adapt to the normal appearance of the scene. Change is detected as pixel intensities having low probability density according to the mixture distribution. It is the case that such change objects will introduce new modes into the mixture. However, these modes will typically have low weight compared to appearance mixture components corresponding to the stable scene background.

The probability of change can be computed as a Bayesian posterior based on the scene background mixture. That is,

$$P(\text{change}|x) = \frac{P(\text{change})}{P(\text{change}) + p(x|\text{background})(1 - P(\text{change}))}. \quad (6)$$

Here x is the observed intensity and $p(x|\text{background})$ is the Gaussian mixture learned from the sequence of images. The probability $P(\text{change})$ is the prior belief that the pixel exhibits change, and $P(\text{change}|x)$ is the posterior belief after observing the current image intensity value. This result is based on the assumption that image intensities are normalized to the range $[0, 1]$ and the probability density for foreground (change) intensities is uniform, i.e., $p(x|\text{change}) = 1$. Change is detected for pixels having $P(\text{change}|x)$ greater than a threshold.

that may or may not be of interest remains a significant challenge.

Experimental Results

An example of detected changes using the joint probability method is shown in Fig. 2. Note that significantly different appearance of the orchard on the right and different overall image contrast does not result in change since the differences are accumulated into the joint probability distribution. The method does not completely account for specular reflections of some of the building roofs (e.g., center and lower left) since these occur relatively infrequently. However, the cars in the roadway at the center are detected with reasonable accuracy.

An example of change detection using the Gaussian mixture model is shown in Fig. 3. In this example a sequence of aerial video frames are registered to a common ground plane and provide updates to the Gaussian mixture model at each pixel. Some false change probability can be seen at occluding boundaries and for metal building roofs that cause large variation in image intensity as the viewpoint changes. The actual changes in the scene are the moving vehicles on the roadways, which exhibit highly salient change probability values.

Open Problems

In spite of considerable research on change detection, the current state-of-the-art algorithms only perform well in scenes where appearance is highly consistent and viewpoint variations can be eliminated by using a fixed camera or by direct comparison of 3-d data. One possible way forward is the work of Pollard [5] where Gaussian appearance models are maintained in a full 3-d volumetric grid. This representation can account for variable viewpoint but is still limited to fixed or slowly varying illumination. Even if the viewpoint and illumination issues are overcome, accounting for the vast diversity of types of change



Change Detection, Fig. 2 Two satellite images taken at different times. (a) May 2006. (b) November 2006. (c) The resulting change map from the joint probability algorithm.

Bright intensities indicate high values of $c(i, j)$. The result is by Pollard [5] (Images copyright Digital Globe)



Change Detection, Fig. 3 An example of change probability computed using a Gaussian background mixture model. (a) A typical video frame. (b) The value of $P(\text{change}|x)$ displayed with white = 1

References

1. Congalton R, Green K (2009) Assessing the accuracy of remotely sensed data: principles and practices, 2nd edn. CRC Press, Boca Raton, FL
2. Stauffer C, Grimson W (1999) Adaptive background mixture models for real-time tracking. In: Proceedings of the international conference on computer vision and pattern recognition (CVPR), Fort Collins, Colorado, New York, vol 2, pp 246–252
3. Girardeau-Montau D, Roux M, Marc R, Thibault G (2005) Change detection on points cloud data acquired with a ground laser scanner. Remote sensing and spatial information sciences 36 (part3/W19), Enschede, The Netherland, pp 30–35
4. Carlotto MJ (2005) Detection and analysis of change in remotely sensed imagery with application to wide area

surveillance. IEEE trans on image process, New York, 2(3), pp 189–202

5. Pollard T (2009) Comprehensive 3-d change detection using volumetric appearance modeling. Brown University, Providence, RI

Chromaticity

Michael H. Brill
Datacolor, Lawrenceville, NJ, USA

Related Concepts

- [Trichromatic Theory](#)

Definition

Chromaticity is a representation of the tristimulus values of a light (see entry on ► [Trichromatic Theory](#)) by only two numbers, computed so as to suppress via ratios the absolute intensity of the light. The two numbers (called chromaticity coordinates) define a space (called chromaticity space) in which any additive mixture of two lights lies on a straight line between those two lights.

Theory

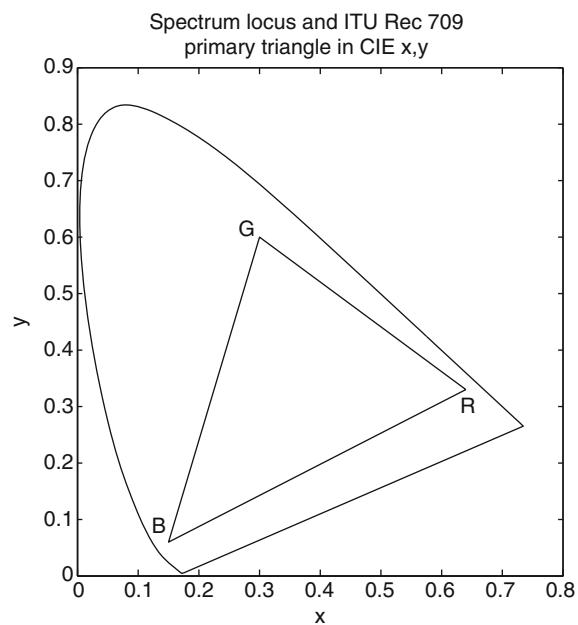
Most often, chromaticity is derived from a standardized tristimulus coordinate system representing human vision (e.g., a CIE XYZ system [1, 2]). If a light has tristimulus values X, Y, and Z, its chromaticity coordinates are conventionally defined as the ratios

$x = X/(X + Y + Z)$ and $y = Y/(X + Y + Z)$. More generally, the chromaticity coordinates could be defined as the ratios $q_1 = (aX + bY + cZ)/(pX + qY + rZ)$ and $q_2 = (dX + eY + fZ)/(pX + qY + rZ)$, where $a, b, c, d, e, f, p, q, r$ are constants that do not depend on the light. The above mapping from (X, Y, Z) to (q_1, q_2) is called a homogeneous central projection. It has the property of showing straight lines in (X, Y, Z) as straight lines in (q_1, q_2) . Hence it is obvious from looking at the chromaticity coordinates of three lights whether one of the lights could have been additively mixed from the other two. This property is useful for visualizing the color gamut (set of producible colors) of a self-luminous trichromatic display such as a cathode-ray tube (CRT) or a liquid-crystal display (LCD). The primaries (say, red, green, and blue) for such displays define three points in chromaticity space, and the triangle they generate spans the chromaticities producible by the display.

To illustrate the geometry of chromaticity, the figure below shows the 1931 CIE (x, y) diagram (Fig. 1). The horseshoe-shaped curve (from 380 nm at the left-hand end to 720 nm on the right-hand end) is the set of all monochromatic lights within the space, and is called the spectrum locus. The straight line connecting the ends of the curve is the line of purples. (There are no monochromatic purples.) Finally, the vertices of the triangle inside the spectrum locus represent the R, G, and B display primaries prescribed by ITU Recommendation 709. This set of primaries is only one example of many prescribed by standards bodies [3], and of even more sets manifested in real displays.

Various useful constructions are possible in chromaticity space. For example, the dominant wavelength of a light is defined by the point on the spectrum locus intercepted by a ray from the agreed-upon white chromaticity through the chromaticity of the light in question. Another example is the set of lights defined by the blackbody radiators (parameterized by their temperature). These lights collectively define the blackbody locus, which of course lies within the spectrum locus. The set of conventional daylights also forms a curve that is close to the blackbody locus. As another example, it is possible to transform from one tristimulus primary set to another such set using only information in the chromaticity domain [4].

Because a digital camera is a trichromatic device, one can also think of a camera's response to light as having a chromaticity. Such chromaticities are useful because they suppress spatial variations of light



Chromaticity, Fig. 1 CIE (x, y) chromaticity diagram with spectrum locus and RGB display primaries

intensity in an image (e.g., at shadow boundaries) and thereby facilitate image segmentation by object color (see entry on Band Ratios). Of course, band-ratio pairs are more general than chromaticities because the ratios can be separately defined without the mixture-on-a-straight-line constraint required for a chromaticity.

It should be noted that camera chromaticity is not at all the same as the human-vision chromaticity and must not be confused with it. In fact, two lights that have the same human-vision chromaticity can have different camera chromaticities, and vice versa. This effect is related to metamerism as noted in the entry on Trichromatic Theory. It should also be noted that, for the same reason, camera-derived values I, H, S used by computer-vision and image-processing applications are also not transformable to human perceptual attributes (e.g., intensity, hue, and saturation).

For both cameras and humans, chromaticity generalizes to a non-trichromatic system (i.e., one that has a number of different sensor types that is different from three). A sensor system with N sensor types delivers N -stimulus values for each light, the chromaticity space has $N-1$ dimensions (so as to suppress the light intensity), and the chromaticity coordinates comprise a homogeneous central projection out of the space of N -stimulus values.

References

1. Wyszecki G, Stiles WS (1982) Color science: concepts and methods, quantitative data and formulae, Chaps. 3 and 5, 2nd edn. Wiley, New York
2. Fairman HS, Brill MH, Hemmendinger H (1997) How the CIE 1931 color-matching functions were derived from Wright-Guild data. *Color Res Appl* 22:11–23
3. Poynton CA (1996) A technical introduction to digital video, Chap. 7. Wiley, New York
4. Brill MH (2008) Transformation of primaries using four chromaticity points and their matches. *Color Res Appl* 33:506–508
5. Fairchild M (2005) Color appearance models, Chap. 3, 2nd edn. Wiley, Chichester
6. MacAdam DL (1985) Color measurement: theme and variations. Springer, Berlin/New York
7. Koenderink JJ (2010) Color for the sciences, Chap. 4. MIT, Cambridge
8. Berns RS (2000) Billmeyer and Saltzman's principles of color technology, 3rd edn. Wiley, New York
9. Hunt RWG (1998) The measurement of colour, 3rd edn. Fountain Press, Kingston-Upon-Thames
10. Hunt RWG (2004) The reproduction of colour, 6th edn. Wiley, Chichester
11. Wandell BA (1995) Foundations of vision. Sinauer, Sunderland

CIE Standard Illuminant

► [Standard Illuminants](#)

Clifford Algebra

► [Geometric Algebra](#)

Clipping

► [Saturation \(Imaging\)](#)

Cluster Sampling

► [Swendsen-Wang Algorithm](#)

Coefficient Rule

► [von Kries Hypothesis](#)

Cognitive Agent

► [Cognitive System](#)

Cognitive System

David Vernon

Informatics Research Centre, University of Skövde, Skövde, Sweden

Synonyms

[Cognitive agent](#)

Related Concepts

► [Cognitive Vision](#)

Definition

A cognitive system is an autonomous system that can perceive its environment, learn from experience, anticipate the outcome of events, act to pursue goals, and adapt to changing circumstances.

Background

There are several scientific perspectives on the nature of cognition and on how it should be modeled. All fall under the general umbrella of cognitive science which embraces the disciplines of neuroscience, artificial intelligence, cognitive psychology, linguistics, and epistemology. Among these differing perspectives, however, there are two broad classes: the *cognitivist* approach based on symbolic information processing representational systems, and the *emergent systems* approach, encompassing connectionist systems, dynamical systems, and enactive systems, all based to a lesser or greater extent on principles of self-organization [1–4]. A third class – *hybrid systems* – attempts to combine something from each of the cognitivist and emergent paradigms. All three approaches have their origins in cybernetics [5] which in the decade from 1943 to 1953 made the first efforts to



formalize what had up to that point been purely psychological and philosophical treatments of cognition. The intention of the early cyberneticians was to create a science of mind, based on logic. Examples of the application of cybernetics to cognition include the seminal paper by McCulloch and Pitts “A logical calculus immanent in nervous activity” [6] and Ashby’s “Design for a Brain” [7].

Theory

The initial attempt in cybernetics to create a science of cognition was followed by the development of an approach referred to as *cognitivism*. The birth of the cognitivist paradigm, and its sister discipline of Artificial Intelligence, dates from a conference held at Dartmouth College, New Hampshire, in July and August 1956 and attended by people such as John McCarthy, Marvin Minsky, Allen Newell, Herbert Simon, and Claude Shannon. Cognitivism holds that cognition is achieved by computation performed on internal symbolic knowledge representations in a process whereby information about the world is abstracted by perception, and represented using some appropriate symbolic data-structure, reasoned about, and then used to plan and act in the world. The approach has also been labeled by many as the information processing or symbol manipulation approach to cognition [1, 8–10]. In most cognitivist approaches concerned with the creation of artificial cognitive systems, the symbolic representations are the descriptive product of a human designer. This is significant because it means that they can be directly accessed and interpreted by humans and that semantic knowledge can be embedded directly into and extracted directly from the system. In cognitivism, the goal of cognition is to reason symbolically about these representations in order to effect the required adaptive, anticipatory, goal-directed behavior. Typically, this approach to cognition will deploy machine learning and probabilistic modeling in an attempt to deal with the inherently uncertain, time-varying, and incomplete nature of the sensory data that is used to drive this representational framework. Significantly, in the cognitivist paradigm, the instantiation of the computational model of cognition is inconsequential: any physical platform that supports the performance of the required symbolic computations will suffice [8]. This principled separation

of operation from instantiation is referred to as functionalism.

In the emergent paradigm, cognition is the process whereby an autonomous system becomes viable and effective in its environment. It does so through a process of self-organization by which the system continually maintains its operational identity through the moderation of mutual system-environment interaction. In other words, the ultimate goal of an emergent cognitive system is to maintain its own autonomy. In achieving this, the cognitive process determines what is real and meaningful for the system: the system constructs its reality – its world and the meaning of its perceptions and actions – as a result of its operation in that world. Consequently, the system’s understanding of its world is inherently specific to the form of the system’s embodiment and is dependent on the system’s history of interactions, i.e., its experiences. This mutual-specification of the system’s reality by the system and its environment is referred to as co-determination [11] and is related to the concept of radical constructivism [12]. This process of making sense of its environmental interactions is one of the foundations of the enactive approach to cognition [13]. Cognition is also the means by which the system compensates for the immediate nature of perception, allowing it to anticipate environmental interaction that occurs over longer time scales, i.e., cognition is intrinsically linked with the ability of an agent to act prospectively: to deal with what might be, not just with what is. Many emergent approaches adhere to the principle that the primary model for cognitive learning is anticipative skill construction rather than knowledge acquisition. Thus, processes which guide action and improve the capacity to guide action form the root capacity of all intelligent systems [14].

As noted already, the emergent paradigm embraces connectionist systems, dynamical systems, and enactive systems. Connectionist systems rely on parallel processing of non-symbolic distributed activation patterns using statistical properties, rather than logical rules, to process information and achieve effective behavior [15]. In this sense, the neural network instantiations of the connectionist model are dynamical systems that capture the statistical regularities in training data [16]. Dynamical systems theory has been used to complement classical approaches in artificial intelligence [17] and it has also been deployed to model natural and artificial cognitive systems [10, 18, 19]. Although dynamical systems theory

approaches often differ from connectionist systems on several fronts, it is better perhaps to consider them complementary ways of describing cognitive systems, dynamical systems addressing macroscopic behavior at an emergent level, and connectionist systems addressing microscopic behavior at a mechanistic level [20]. Enactive systems take the emergent paradigm even further. Enaction [13, 21–23] asserts that cognition is a process whereby the issues that are important for the continued existence of a cognitive entity are brought out or enacted: co-determined by the entity and the environment in which it is embedded. Thus, enaction entails that a cognitive system operates autonomously, that it generates its own models of how the world works, and that the purpose of these models is to preserve the system's autonomy.

Considerable effort has gone into developing hybrid approaches which combine aspects of cognitivist and emergent systems. Typically, hybrid systems exploit symbolic knowledge to represent the agent's world and logical rule-based systems to reason about this knowledge in order to achieve goals and select actions, while at the same time using emergent models of perception and action to explore the world and construct this knowledge. Thus, hybrid systems still use cognitivist representations and representational invariances but they are constructed by the system itself as it interacts with and explores the world rather than through a priori specification or programming. Consequently, as with emergent systems, the agent's ability to understand the external world is dependent on its ability to interact flexibly with it, and interaction is the organizing mechanism that establishes the association between perception and action.

Cognitivism and artificial intelligence research are strongly related. In particular, Newell and Simon's "Physical Symbol System" approach to artificial intelligence [8] has been extremely influential in shaping how we think about intelligence, natural as well as computational. In their 1976 paper, two hypotheses are presented: the *Physical Symbol System Hypothesis* and the *Heuristic Search Hypothesis*. The first hypothesis is that a physical symbol system has the necessary and sufficient means for general intelligent action. This implies that any system that exhibits general intelligence is a physical symbol system and any physical symbol system of sufficient size can be configured to exhibit general intelligence. The second hypothesis states that the solutions to problems are represented as

symbol structures and that a physical-symbol system exercises its intelligence in problem-solving by search, i.e., by generating and progressively modifying symbol structures in an effective and efficient manner until it produces a solution structure. This amounts to an assertion that symbol systems solve problems by heuristic search, i.e., the successive generation of potential solution structures. The task of intelligence, then, is to avert the ever-present threat of the exponential explosion of search. Subsequently, Newell defined intelligence as the degree to which a system approximates the ideal of a knowledge-level system [24]. A knowledge-level system is one which can bring to bear *all* its knowledge onto *every* problem it attempts to solve (or, equivalently, every goal it attempts to achieve). Perfect intelligence implies complete utilization of knowledge. It brings this knowledge to bear according to the *principle of maximum rationality* which was proposed by Newell in 1982 [25] as follows: "If an agent has knowledge that one of its actions will lead to one of its goals, then the agent will select that action." Anderson [26] later offered a slightly different principle, the *principle of rationality*, sometimes referred to as rational analysis, stated as follows: "the cognitive system optimizes the adaptation of the behavior of the organism." Note that Anderson's principle considers optimality to be necessary for rationality, something that Newell's principle does not.

Cognitivist and emergent approaches are normally contrasted on the basis of the symbolic or non-symbolic nature of their computational operation and representational framework. Cognitivist systems typically use production systems to effect rule-based manipulation of symbol tokens whereas emergent systems exploit dynamical processes of self-organization in which representations are encoded in global system states. However, the distinction between cognitivist and emergent is not restricted to the issue of symbolic representation and they can be contrasted on the basis of several other characteristics such as semantic grounding, temporal constraints, inter-agent epistemology, embodiment, perception, action, anticipation, adaptation, motivation, autonomy, among others [27].

The differences between the cognitivist and the emergent paradigm can be traced to their underlying distinct philosophies [28]. Broadly speaking, cognitivism is dualist, functionalist, and positivist. It is dualist in the sense that there is a fundamental distinction between the mind (the computational processes) and



the body (the computational infrastructure and, if required, the physical structure that instantiates any physical interaction). It is functionalist in the sense that the actual instantiation and computational infrastructure is inconsequential: any instantiation that supports the symbolic processing is sufficient. It is positivist in the sense that they assert a unique and absolute empirically-accessible external reality that is apprehended by the senses and reasoned about by the cognitive processes. In contrast, emergent systems are neither dualist nor functionalist, since the system's embodiment is an intrinsic component of the cognitive process, nor positivist, since the form and meaning of the system's world is dependent in part on the system itself. The emergent paradigm, and especially the enactive approach, can trace its roots to the philosophy of phenomenology [28, 29].

A criticism often leveled at cognitivist systems is that they are relatively poor at functioning effectively outside well-defined problem domains because they tend to depend on in-built assumptions and embedded knowledge arising from design decisions. Emergent systems should in theory be much less brittle because they develop through mutual specification and co-determination with the environment. However, the ability to build artificial cognitive systems based on emergent principles is very limited at present, and cognitivist and hybrid systems currently have more advanced capabilities within a narrower application domain.

Any cognitive system is inevitably going to be complex. Nonetheless, it is also the case that it will exhibit some degree of structure. This structure is often encapsulated in what is known as a cognitive architecture [30]. Although used freely by proponents of the cognitivist, emergent, and hybrid approaches to cognitive systems, the term “cognitive architecture” originated with the seminal cognitivist work of Newell et al. [25]. Consequently, the term has a very specific meaning in this paradigm where cognitive architectures represent attempts to create unified theories of cognition [24, 31], i.e., theories that cover a broad range of cognitive issues, such as attention, memory, problem-solving, decision-making, learning, from several aspects including psychology, neuroscience, and computer science. In the cognitivist paradigm, the focus of a cognitive architecture is on the aspects of cognition that are constant over time and that are independent of the task. Since cognitive

architectures represent the fixed part of cognition, they cannot accomplish anything in their own right and need to be provided with or acquire knowledge to perform any given task. For emergent approaches to cognition, which focus on development from a primitive state to a fully cognitive state over the lifetime of the system, the architecture of the system is equivalent to its phylogenetic configuration: the initial state from which it subsequently develops through ontogenesis.

Open Problems

The study of cognitive systems is a maturing discipline with contrasting approaches. Consequently, there are several open problems. These include the role of physical embodiment, the need for development, the system's cognitive architecture, the degree of autonomy required, the issue of symbol grounding, the problem of goal specification, the ability to explain the rationale for selection actions, the problem of generating generalized concepts and transferring knowledge from one context to another, and the interdependence of perception and action. The nature of any resolution of these problems is inextricably linked to the choice of paradigm: cognitivist, emergent, or hybrid.

The role of physical embodiment in a cognitive system [32–34] depends strongly on the chosen paradigm. Due to their functionalist characteristics, cognitivist systems do not depend on physical embodiment to operate successfully but there is nothing to prevent them from being embodied if that is what the task in hand requires. Emergent systems, by definition, require embodiment since the body plays a key role in the way a cognitive system comes to understand – make sense of – its environment. If a body is required, the form of embodiment must still be specified [35]. This is significant because, in the emergent paradigm at least, the ability of two cognitive agents to communicate effectively requires them to have similar embodiments so that they have a shared history of interaction and a common epistemology.

The extent to which a cognitive system requires a capacity for development and, if so, the mechanisms by which development can take place are both open problems. In natural systems, growth is normally associated with development. However, growth in artificial systems remains a distant goal, although one whose

achievement would open up many avenues of fruitful enquiry in cognitive systems. For current state-of-the-art cognitive systems, one can define development as the process by which a system discovers for itself the models that characterize its interactions with its environment. This contrasts with learning as the process whereby the parameters of an existing model are estimated or improved. Development, then, requires a capacity for self-modification [36] and in embodied emergent systems leads to an increased repertoire of effective actions and a greater ability to anticipate the need for and outcome of future actions [27].

The capacity to develop introduces another open issue: the minimal phylogenetic configuration – the perceptual, cognitive, and motoric capabilities with which a system is endowed at “birth” – that is required to facilitate subsequent ontogenesis – development and learning through exploration and social interaction [27]. This issue is related to the specification of the system’s cognitive architecture and the necessary and sufficient conditions that must be satisfied for cognitive behavior to occur in a system. In addressing these issues, there is a trade-off between the initial phylogeny and the potential for subsequent development. This trade-off is reflected by the existence of two types of species in nature: precocial and altricial. Precocial species are those that are born with well-developed behaviors, skills, and abilities which are the direct result of their genetic make-up (i.e., their phylogenetic configuration). As a result, precocial species tend to be quite independent at birth. Altricial species, on the other hand, are born with poor or undeveloped behaviors and skills, and are highly dependent for support. However, in contrast to precocial species, they proceed to learn complex cognitive skills over their lifetime (i.e., through ontogenetic development). The precocial and the altricial effectively define a spectrum of possible configurations of phylogenetic configuration and ontogenetic potential [37]. The problem is to identify a feasible point in this spectrum that will yield a cognitive system capable of developing the skills we require of it.

Autonomy is a crucial issue for cognitive systems [38] but the degree of autonomy required is unclear. To an extent, it depends on how autonomy is defined and which paradigm of cognition is being considered. Definitions range from self-regulation and homeostasis to the ability of a system to contribute to its own persistence [39]. In the former case, self-regulation is

often cast as a form of self-control so that the systems can operate without interference from some outside agent, such as a human user. In the latter case, autonomy is the self-maintaining organizational feature of living creatures that enables them to use their own capacities to manage their interactions with the world in order to remain viable [14]. Cognitivist systems tend to adopt the former definition, emergent systems, the latter.

Broadly speaking, cognitivist systems exploit symbolic representations while emergent systems exploit sub-symbolic state-based representations, with hybrid systems using both. The manner in which cognitivist and hybrids systems ground their symbolic representations in experience is still an open issue [40], with some arguing for a bottom-up approach [41] and others for a process of learned association, where meaning is attached rather than grounded [37].

The opening definition of a cognitive system states that it can act to achieve goals. The specification of these goals poses a significant challenge due to the autonomous nature of cognitive systems. It is more easily resolved for cognitivist systems since the goals can be hard-wired into the cognitive architecture. It is less clear how goals can be specified in an emergent system since the over-arching goal here is the maintenance of the system’s autonomy. The goals of such a system reflect its intrinsic motivations and its associated value system [42]. The problem is to understand how to engineer this value system to ensure that the system is motivated to act in a way that satisfies goals which are external to the system and to decide how these goals can be communicated to the system.

Ideally, in addition to the characteristics of a cognitive system listed in the opening definition – autonomy, perception, learning, anticipation, goal-directed action, and adaptation – a cognitive computer system should also be able to say what it is doing and why it is doing it, i.e., it should be able to explain the reasons for an action [43]. This would enable the system to identify potential problems which might appear when carrying out a task and it would know when it needed new information in order to complete that task. Consequently, a cognitive system would be able to view a problem in several different ways and to look at different alternative ways of tackling it. In a sense, this is something similar to the issue discussed above about cognition involving an ability to anticipate the need for actions and their outcome. The difference in



this case is that the cognitive system is considering not just one but many possible sets of needs and outcomes. In a sense, it is adapting *before* things do not go according to plan. From this point of view, cognition also involves a sense of self-reflection.

Cognitive systems also learn from experience and adapt to changing circumstances. To do this, the system must have some capacity for generalization so that concepts can be formed from specific instances and so that knowledge and know-how can be transferred from one context to another. This capacity would allow the system to adapt to new application scenarios and to explore the hypothetical situations that arise from the self-reflection mentioned above. It is unclear at present how such generalized conceptual knowledge and know-how should be generated, represented, and incorporated into the system dynamics.

Perception and action have been demonstrated to be co-dependent in biological systems. Perceptual development depends on what actions an infant is capable of and what use objects and events afford in the light of these capabilities. This idea of the action-dependent perceptual interpretation of an object is referred to as its affordance [44]. In neuroscience, the tight relationship between action and perception is exemplified by the presence of mirror neurons, neurons that become active when an action is performed and when the action or a similar action is observed being performed by another agent. It is significant that these neurons are specific to the goal of the action and not the mechanics of carrying it out. The related Ideomotor Theory [45] asserts the existence of such a common or co-joint representational framework for perception and action. Such a framework would facilitate the inference of intention and the anticipation of an outcome of an event due to the goal-oriented nature of the action. The realization of an effective co-joint perception-action framework remains an important challenge for cognitivist and emergent approaches alike.

Although clearly there are some fundamental differences between the cognitivist and the emergent paradigms, the gap between the two shows some signs of narrowing. This is mainly due to (1) a recent movement on the part of proponents of the cognitivist paradigm to assert the fundamentally important role played by action and perception in the realization of a cognitive system [32]; (2) the move away from the view that internal symbolic representations are the only valid form of representation [2]; and (3)

the weakening of the dependence on embedded a priori knowledge and the attendant-increased reliance on machine learning and statistical frameworks both for tuning system parameters and the acquisition of new knowledge. This suggests that hybrid approaches may be the way forward, especially if a principled synthesis of cognitivist and emergent approaches is possible, such as “dynamic computationalism” [2] or “computational mechanics” [46]. Hybrid approaches appear to many to offer the best of both worlds – the adaptability of emergent systems and the advanced starting point of cognitivist systems – since the representational invariances and representational frameworks need not be learned but can be designed in and since the system populates these representational frameworks through learning and experience. However, it is uncertain that one can successfully combine what are ultimately highly incompatible underlying philosophies. Opinion is divided, with arguments both for (e.g., [2, 40, 46]) and against (e.g., [47]).

References

1. Varela FJ (1992) Whence perceptual meaning? A cartography of current ideas. In: Varela FJ, Dupuy JP (eds) Understanding origins – contemporary views on the origin of life, mind and society. Boston studies in the philosophy of science. Kluwer Academic Publishers, Dordrecht, pp 235–263
2. Clark A (2001) Mindware – an introduction to the philosophy of cognitive science. Oxford University Press, New York
3. Freeman WJ, Núñez R (1999) Restoring to cognition the forgotten primacy of action, intention and emotion. *J Conscious Stud* 6(11–12):ix–xix
4. Winograd T, Flores F (1986) Understanding computers and cognition – a new foundation for design. Addison-Wesley Publishing Company, Inc., Reading
5. Ross Ashby W (1957) An introduction to cybernetics. Chapman and Hall, London
6. McCulloch WS, Pitts W (1943) A logical calculus of ideas immanent in nervous activity. *Bull Math Biophys* 5:115–133
7. Ross Ashby W (1954) Design for a brain. Chapman and Hall, London
8. Newell A, Simon HA (1976) Computer science as empirical inquiry: symbols and search. *Commun Assoc Comput Mach* 19:113–126. Tenth Turing award lecture, ACM, 1975
9. Haugland J (ed) (1997) Mind design II: philosophy, psychology, artificial intelligence. MIT, Cambridge, MA
10. Kelso JAS (1995) Dynamic patterns – the self-organization of brain and behaviour, 3rd edn. MIT, Cambridge, MA
11. Maturana H, Varela F (1987) The tree of knowledge – the biological roots of human understanding. New Science Library, Boston & London

12. von Glaserfeld E (1995) Radical constructivism. RoutledgeFalmer, London
13. Stewart J, Gapenne O, Di Paolo EA (2011) Enaction: toward a new paradigm for cognitive science. MIT, Cambridge, MA
14. Christensen WD, Hooker CA (2000) An interactivist-constructivist approach to intelligence: self-directed anticipative learning. *Philos Psychol* 13(1):5–45
15. Medler DA (1998) A brief history of connectionism. *Neural Comput Surv* 1:61–101
16. Smolensky P (1996) Computational, dynamical, and statistical perspectives on the processing and learning problems in neural network theory. In: Smolensky P, Mozer MC, Rumelhart DE (eds) Mathematical perspectives on neural networks. Erlbaum, Mahwah, NJ, pp 1–15
17. Reiter R (2001) Knowledge in action: logical foundations for specifying and implementing dynamical systems. MIT, Cambridge, MA
18. Thelen E, Smith LB (1994) A dynamic systems approach to the development of cognition and action. MIT press/Bradford books series in cognitive psychology. MIT, Cambridge, MA
19. Port RF, van Gelder T (1995) Mind as motion – explorations in the dynamics of cognition. Bradford Books, MIT, Cambridge, MA
20. McClelland JL, Vallabha G (2006) Connectionist models of development: mechanistic dynamical models with emergent dynamical properties. In: Spencer JP, Thomas MSC, McClelland JL (eds) Toward a new grand theory of development? Connectionism and dynamic systems theory re-considered. Oxford University Press, New York
21. Maturana HR, Varela FJ (1980) Autopoiesis and cognition – the realization of the living. Boston studies on the philosophy of science. D. Reidel Publishing Company, Dordrecht
22. Varela F, Thompson E, Rosch E (1991) The embodied mind. MIT, Cambridge, MA
23. Vernon D (2010) Enaction as a conceptual framework for development in cognitive robotics. *Paladyn J Behav Robot* 1(2):89–98
24. Newell A (1990) Unified theories of cognition. Harvard University Press, Cambridge, MA
25. Newell A (1982) The knowledge level. *Artif Intell* 18(1): 87–127
26. Anderson J (1999) Cognitive architectures in rational analysis. In: van Lehn K (ed) Architectures for intelligence. Lawrence Erlbaum Associates, Hillsdale, pp 1–24
27. Vernon D, von Hofsten C, Fadiga L (2010) A roadmap for cognitive development in humanoid Robots. Volume 11 of cognitive systems monographs (COSMOS). Springer, Berlin
28. Vernon D, Furlong D (2007) Philosophical foundations of enactive AI. In: Lungarella M, Iida F, Bongard JC, Pfeifer R (eds) 50 years of AI. Volume LNAI 4850. Springer, Heidelberg, pp 53–62
29. Froese T, Ziemke T (2009) Enactive artificial intelligence: investigating the systemic organization of life and mind. *Artif Intell* 173:466–500
30. Langley P, Laird JE, Rogers S (2009) Cognitive architectures: research issues and challenges. *Cogn Syst Res* 10(2):141–160
31. Anderson JR, Bothell D, Byrne MD, Douglass S, Lebiere C, Qin Y (2004) An integrated theory of the mind. *Psychol Rev* 111(4):1036–1060
32. Anderson ML (2003) Embodied cognition: a field guide. *Artif Intell* 149(1):91–130
33. Steels L (2007) Fifty years of AI: from symbols to embodiment – and back. In: Lungarella M, Iida F, Bongard JC, Pfeifer R (eds) 50 years of AI. Volume LNAI 4850. Springer, Heidelberg, pp 18–28
34. Vernon D (2008) Cognitive vision: the case for embodied perception. *Image Vis Comput* 26(1):127–141
35. Ziemke T (2003) What's that thing called embodiment? In: Alterman R, Kirsh D (eds) Proceedings of the 25th annual conference of the cognitive science society. Lund university cognitive studies. Lawrence Erlbaum, Mahwah, pp 1134–1139
36. Weng J (2004) Developmental robotics: theory and experiments. *Int J Humanoid Robot* 1(2):199–236
37. Sloman A, Chappell J (2005) The altricial-preocial spectrum for robots. In: IJCAI '05 – 19th international joint conference on artificial intelligence, Edinburgh
38. Varela F (1979) Principles of biological autonomy. Elsevier North Holland, New York
39. Bickhard MH (2000) Autonomy, function, and representation. *Artif Intell Spec Issue Commun Cogn* 17(3–4): 111–131
40. Barsalou LW (2010) Grounded cognition: past, present, and future. *Top Cogn Sci* 2:716–724
41. Harnad S (1990) The symbol grounding problem. *Phys D* 42:335–346
42. Edelman GM (2006) Second nature: brain science and human knowledge. Yale University Press, New Haven and London
43. Brachman RJ (2002) Systems that know what they're doing. *IEEE Intell Syst* 17(6):67–71
44. Gibson JJ (1979) The ecological approach to visual perception. Houghton Mifflin, Boston
45. Stock A, Stock C (2004) A short history of ideo-motor action. *Psychol Res* 68(2–3):176–188
46. Crutchfield JP (1998) Dynamical embodiment of computation in cognitive processes. *Behav Brain Sci* 21(5):635–637
47. Christensen WD, Hooker CA (2004) Representation and the meaning of life. In: Clapin H, Staines P, Slezak P (eds) Representation in mind: new approaches to mental representation. Elsevier, Oxford, pp 41–70

Cognitive Vision

David Vernon

Informatics Research Centre, University of Skövde,
Skövde, Sweden

Related Concepts

►Cognitive System; ►Visual Cognition



Definition

Cognitive vision refers to computer vision systems that can pursue goals, adapt to unexpected changes of the visual environment, and anticipate the occurrence of objects or events.

Background

The field of cognitive vision grew from the broader area of computer vision in response to a need for vision systems that are more widely applicable, that are able adapt to novel scenes and tasks, that are robust to unexpected variations in operating conditions, and that are fast enough to deal with the timing requirements of these tasks [1]. Adaptability entails the ability to acquire knowledge about the application domain, thereby removing the need to embed all the required knowledge in the system when it is designed. Robustness allows the system to be tolerant to changes in environmental conditions so that system performance is not negatively impacted by them when carrying out a given task. Speed and the ability to pay attention to critical events are essential when providing feedback to users and devices in situations which change unexpectedly [2, 18]

While computer vision systems routinely address signal processing of sensory data and reconstruction of 3D scene geometry, cognitive vision goes beyond this by providing a capability for conceptual characterization of the scene structure and dynamics using qualitative representations. Having knowledge about the scene available in conceptual form allows the incorporation of consistency checks through the use of, e.g., logic inference engines. These checks can be applied both to the knowledge that is embedded in the system at the outset and the knowledge that the system learns for itself. Consistency checking applies across several scales of space and time, requiring cognitive vision to have an ability to operate with past, present, and future events. These consistency checks are one way in which the robustness associated with cognitive vision can be achieved [3]. Furthermore, the conceptual knowledge generated by cognitive vision can, if required, be communicated to a human user in natural language [4]. This linguistic communication is one manifestation of an autonomous system

demonstrating its understanding of the visual events in its environment [3].

Theory

Cognitive vision entails abilities to anticipate future events and to interpret a visual scene in the absence of complete information. To achieve this, a cognitive system must have the capacity to acquire new knowledge and to use it to fill in gaps that are present in what is being made immediately available by the visual sensors: to extrapolate in time and space to achieve a more robust and effective understanding of the underlying behavior of the sensed world. In the process, the system learns, anticipates, and adapts. These three characteristics of learning, anticipation, and adaptivity are the hallmarks of cognition, in general, and cognitive vision, in particular [6, 7].

A key property of cognitive vision is its capacity to exhibit a robust performance even in scenarios that were not anticipated when it was designed. The degree to which a system can deal with unexpected circumstances will vary. Systems that can adapt autonomously to arbitrary situations are unrealistic at present but it is plausible that they should be able to deal with new variants of visual form, function, and behavior, and also incremental changes in context. Ideally, a cognitive vision system should be able to recognize and adapt to novel variations in the current visual environment, generalize to new contexts and application domains, interpret and predict the behavior of agents detected in the system's environment, and communicate an understanding of the environment to other systems, including humans.

A cognitive vision system is a visually enabled cognitive system, defined in this encyclopedia as “an autonomous system that can perceive its environment, learn from experience, anticipate the outcome of events, act to pursue goals, and adapt to changing circumstances.” Since cognitive vision is principally a mode of perception, physical action – with the possible exception of the camera movements associated with active vision – usually falls outside its scope. However, speech acts may be involved when communicating an interpretation of the scene in conceptual terms through language [5]. Since cognitive vision is a particular type of cognitive system, all the issues identified in the cognitive system entry in this encyclopedia

apply equally to cognitive vision. They will not be revisited here apart from noting that there are several scientific perspectives on the nature of cognition and on how it should be modeled. Among these differing perspectives, there are two broad classes: the *cognitivist* approach based on symbolic information processing representational systems, and the *emergent systems* approach, encompassing connectionist systems, dynamical systems, and enactive systems, all based to a lesser or greater extent on principles of self-organization. A third class – *hybrid systems* – attempts to combine something from each of the cognitivist and emergent paradigms. The vast majority of cognitive vision systems adopt either a cognitivist or a hybrid approach, with matching cognitive architectures [8].

The term *visual cognition* is strikingly similar to the term *cognitive vision*. However, they are not equivalent. Visual cognition is a branch of cognitive psychology concerned with research on visual perception and cognition in humans [9, 10, 19]. It addresses several distinct areas such as object recognition, face recognition, scene understanding, visual attention (including visual search, change blindness, repetition blindness, and the control of eye movements), short-term and long-term visual memory, and visual imagery. It is also concerned with the representation and recognition of visual information currently being perceived by the senses and with reasoning about memorized visual imagery. Thus, visual cognition addresses many visual mechanisms that are relevant to cognitive vision but without necessarily treating the entire cognitive system or the realization of these mechanisms in artificial systems.

Application

Several applications of cognitive vision may be found in [11–13]. Examples include natural language description of traffic behavior [5], autonomous control of cars [14], and observation and interpretation of human activity [15, 16].

Open Problems

All of the open problems associated with cognitive systems apply equally to cognitive vision. Three which are particularly relevant are highlighted here.

The first concerns embodiment [17]. There is no universal agreement on whether or not a cognitive vision system must be embodied. Even if it is, several forms of embodiment are possible. One form is a physical robot capable of moving in space, manipulating the environment, and experiencing the physical forces associated with that manipulation. Other forms of embodiment do not involve physical contact and simply require the system to be able to change the state of its visual environment, for example, a surveillance system which can control ambient lighting. These alternative forms of embodiment are consistent with the cognitivist and hybrid paradigms of cognition but do not satisfy the requirements of the emergent approach.

Learning in cognitive vision presents several significant challenges. Since cognitive vision systems do not have all the knowledge required to carry out their tasks, they need to learn. More specifically, they need to be able to learn in an incremental, continuous, open-ended, and robust manner, with learning and recognition being interleaved, and with both improving over time. Since the learning process will normally be effected autonomously without supervision, the learning technique needs to be able to distinguish between good and bad data, otherwise bad data may corrupt the representation and cause errors to become embedded and to propagate. Furthermore, the use of learning in several domains is required, including perceptual (spatiotemporal) and conceptual (symbolic) domains, as well as in mapping between them. The mapping from perceptual to conceptual facilitates communication, categorization, and reasoning, whereas the mapping from conceptual to perceptual facilitates contextualization and embodied action. Learning may be interpreted in a restricted sense as the estimation of the parameter values that govern the behavior of models that have been designed into the system, or in a more general sense as the autonomous generation of mappings that represent completely new models.

The identification and achievement of goals in cognitive vision presents a further challenge. With cognitivist approaches, goals are specified explicitly by designers or users in terms of the required outcome of cognitive behavior. With emergent approaches, goals are more difficult to specify since cognitive behavior is an emergent consequence of the system dynamics. Consequently, they have to be specified in terms



of constraints or boundary conditions on the system configuration, either through phylogenetic configuration or ontogenetic development, or both. It is a significant challenge to understand how implicit goals can be specified and incorporated, and how externally communicated goals can be introduced to the system from its environment or from those interacting with it, e.g., through some form of conditioning, training, or communication.

References

1. Christensen HI, Nagel HH (2006) Introductory remarks. In: Christensen HI, Nagel HH (eds) Cognitive vision systems: sampling the spectrum of approaches. Volume 3948 of LNCS. Springer, Heidelberg, pp 1–4
2. Tsotsos JK (2006) Cognitive vision need attention to link sensing with recognition. In: Christensen HI, Nagel HH (eds) Cognitive vision systems: sampling the spectrum of approaches. Volume 3948 of LNCS. Springer, Heidelberg, pp 25–35
3. Nagel HH (2003) Reflections on cognitive vision systems. In: Crowley J, Piater J, Vincze M, Paletta L (eds) Proceedings of the third international conference on computer vision systems, ICVS 2003. Volume LNCS 2626. Springer, Berlin, Heidelberg, pp 34–43
4. Arens M, Ottlik A, Nagel HH (2002) Natural language texts for a cognitive vision system. In: Harmelen FV (ed) Proceedings of the 15th European conference on artificial intelligence (ECAI-2002). IOS Press, Amsterdam, pp 455–459
5. Nagel HH (2004) Steps toward a cognitive vision system. *AI Magazine* 25(2):31–50
6. Auer P et al (2005) A research roadmap of cognitive vision. ECVision: European network for research in cognitive vision systems. <http://www.ecvision.org>
7. Vernon D (2006) The space of cognitive vision. In: Christensen HI, Nagel HH (eds) Cognitive vision systems: sampling the spectrum of approaches. Volume 3948 of LNCS. Springer, Heidelberg, pp 7–24
8. Granlund G (2006) Organization of architectures for cognitive vision systems. In: Christensen HI, Nagel HH (eds) Cognitive vision systems: sampling the spectrum of approaches. Volume 3948 of LNCS. Springer, Heidelberg, pp 37–55
9. Pinker S (1984) Visual cognition: an introduction. *Cognition* 18:1–63
10. Coltheart V (ed) (2010) Tutorials in visual cognition. Macquarie monographs in cognitive science. Psychology Press, London
11. Neumann B (ed) (2005) Künstliche Intelligenz, special issue on cognitive computer vision, vol 2. Böttcher IT Verlag, Bremen
12. Christensen HI, Nagel HH (2006) Cognitive vision systems: sampling the spectrum of approaches. Volume 3948 of LNCS. Springer, Heidelberg
13. Buxton H (ed) (2008) Image vision comput, special issue on cognitive vision 26(1)
14. Dickmanns ED (2004) Dynamic vision-based intelligence. *AI Magazine* 25(2):10–29
15. Sage K, Howell J, Buxton H (2005) Recognition of action, activity and behaviour in the actIPret project. *Künstliche Intell Spec Issue Cogn Comput Vis* 19(2):30–34
16. Crowley JL (2006) Things that see: context-aware multimodal interaction. In: Christensen HI, Nagel HH (eds) Cognitive vision systems: sampling the spectrum of approaches. Volume 3948 of LNCS. Springer, Heidelberg, pp 183–198
17. Vernon D (2008) Cognitive vision: the case for embodied perception. *Image Vis Comput* 26(1):127–141
18. Tsotsos JK (2011) A computational perspective on visual attention. MIT Press, Cambridge
19. Cavanagh P (2011) Visual cognition. *Vision Res* 51(13): 1538–1551

Color Adaptation

► [von Kries Hypothesis](#)

Color Appearance Models

► [Color Spaces](#)

Color Balance

► [White Balance](#)

Color Constancy

Marc Ebner

Institut für Mathematik und Informatik, Ernst Moritz Arndt Universität Greifswald, Greifswald, Germany

Definition

Color constancy is the ability to perceive colors as approximately constant even though the light entering the eye varies with the illuminant. Color constancy also names the field of research investigating the extent of this ability, that is, the conditions under which a color is actually perceived as constant and which factors

influence color constancy. Computer scientists working in the field of color constancy try to mimic this ability in order to produce images which are independent of the illuminant, that is, color constant. Simple color-constancy algorithms, also known under the name automatic white balance, are used in digital cameras to compute a color-corrected output image. The input of a color constancy algorithm is often one image taken under an arbitrary illuminant and the output of a color constancy algorithm is frequently the image as it would appear had it been taken with a canonical illuminant such as CIE Standard Illuminant D65 or a spectrally uniform illuminant.

Background

An introduction to computational color constancy is given by Ebner [1]. Maloney [2] also reviews different algorithms for surface color perception. Color is a product of the brain [3]. When an observer perceives an object, processing starts with the retinal sensors. The sensors in the retina measure the light entering the eye. However, the light entering the eye is dependent on both the spectral characteristics of the illuminant and the reflectance properties of the object. Therefore, without any additional processing, the measured light varies with the illuminant.

In the eye, two different types of retinal sensors exist: rods and cones. The rods are used for viewing when little light is available. The cones are mostly used in bright light conditions for color vision. Three types of cones can be distinguished which absorb light primarily in the red, green, and blue parts of the spectrum. Similarly, a digital sensor often measures the incident light in three different parts of the spectrum and uses red, green, and blue sub-sensors. However, cameras with four sub-sensors, for example, red, green, blue, and cyan also exist.

Suppose that an observer views a diffusely reflecting surface which uniformly reflects the incident light. Now assume that the surface is illuminated by a candle. A candle emits more light toward the red part of the spectrum. The candle light will reach the surface where part of the light will be absorbed and the remainder will be reflected. Part of the reflected light enters the eye where it is measured. The sensitivity function of the sensors in combination with the amount of light entering the eye will determine how strongly the

sensors respond. Now consider another illuminant with a higher color temperature, for example, daylight or an electronic flash. Such an illuminant will emit more light toward the blue spectrum compared to the candle. If the same surface is viewed with an illuminant that has a high color temperature, then the sensors shift their response toward the blue part of the spectrum. Assuming a normalized response of all three sensors, then the reflected light will have a red color cast when the surface is illuminated by a candle. The measured light will appear white during daylight and it will have a bluish color cast for an illuminant with a high color temperature. The observer, however, will be able to call out the correct color of the surface independent of the illuminant. Color constancy algorithms try to mimic this ability by computing an image which is independent of the illuminant.

Theory

Let $I(\lambda, x, y)$ be the irradiance captured by either a digital sensor or by the eye at position (x, y) for wavelength λ . Let $S_i(\lambda)$ be the response function of sensor i . Then the response of the sensor $c_i(x, y)$ at position (x, y) is given by:

$$c_i(x, y) = \int S_i(\lambda) I(\lambda, x, y) d\lambda.$$

The integration is done over all wavelengths to which the sensor responds. Assuming three receptors with sensitivity in the red, green, and blue parts of the spectrum, then $i \in \{r, g, b\}$. In this case, the measurement of the sensor is a three component vector $\mathbf{c} = [c_r, c_g, c_b]$.

The irradiance I falling onto the sensor is a result of the light reflected from an object patch. Let $L(\lambda, x, y)$ be the irradiance falling onto a diffusely reflecting object patch which is imaged at position (x, y) of the sensor arrangement. Let $R(\lambda, x, y)$ be the reflectance of the imaged object patch. Thus,

$$I(\lambda, x, y) = G(x, y) R(\lambda, x, y) L(\lambda, x, y)$$

where $G(x, y)$ is a geometry factor which takes the orientation between the surface and the light source into account. For a diffusely reflecting surface, $G(x, y) = \cos \alpha$ where α is the angle between the unit



vector which points from the surface into the direction of the light source and the normal vector at the corresponding surface position. Thus, the sensor response can be written as:

$$c_i(x, y) = G(x, y) \int S_i(\lambda) R(\lambda, x, y) L(\lambda, x, y) d\lambda.$$

From this equation, it is apparent that the sensor response depends on the orientation of the patch relative to the light source (because of $G(x, y)$); it depends on the sensitivity S_i of the sensor i , the reflectance of the object patch $R(\lambda, x, y)$, and on the illuminant $L(\lambda, x, y)$. Some color constancy algorithms are based on a set of basis functions to model illuminants and reflectances. See Maloney [2] for an introduction.

Color constancy algorithms frequently assume that the sensitivity of the sensors is very narrow band. Assuming that they have the shape of a delta function, $S_i = \delta(\lambda - \lambda_i)$, it holds that:

$$\begin{aligned} c_i(x, y) &= G(x, y) \int \delta(\lambda - \lambda_i) R(\lambda, x, y) L(\lambda, x, y) d\lambda, \\ c_i(x, y) &= G(x, y) R(\lambda_i, x, y) L(\lambda_i, x, y). \end{aligned}$$

This equation is often written in the form:

$$c_i(x, y) = G(x, y) R_i(x, y) L_i(x, y)$$

where the only difference to the previous equation is that the index i is used instead of the parameter λ_i . In this treatment, sensor response, reflectance, and the illuminant is considered only for three distinct wavelengths, that is, color bands, with $i \in \{r, g, b\}$.

A color constancy algorithm tries to discount the illuminant by computing a color constant descriptor $d(\mathbf{c})$ which is independent of the illuminant $\mathbf{L}(x, y) = [L_r(x, y), L_g(x, y), L_b(x, y)]$. Geusebroek et al. [4] have derived several different descriptors which can be computed from \mathbf{c} and are invariant to some imaging conditions such as viewing direction, surface orientation, highlights, illumination direction, illumination intensity, or illumination color. Finlayson and Hordley [5] have shown how a color constant descriptor can be computed provided that the illuminant can be approximated by a black body radiator. Apart from computing a color constant descriptor, a color constancy algorithm usually tries to output an image of

the scene which would either correspond to the perception of a human photographer observing the scene or it would correspond to the image that would have resulted if a spectrally uniform illuminant or illuminant D65 had been used.

An ideal solution to the problem of color constancy would be to compute $\mathbf{R}(x, y) = [R_r(x, y), R_g(x, y), R_b(x, y)]$ from the sensor responses. It is of course clear that this problem cannot be solved without making additional assumptions, because for each position on the sensor array, one only has three measurements but there are seven unknowns (shading, reflectance, and illumination components). Note that the above model for image generation is already a simple model assuming narrow band sensor responses.

A frequently made assumption is that the illuminant varies slowly over the image while reflectance is able to change abruptly between sensor responses. Since color constancy algorithms are based on certain assumptions, they will not work correctly if the assumptions are violated. In many cases, it is possible to find images where the color constancy algorithm does not perform as intended. The goal is to develop algorithms which perform well on most everyday scenes.

Simple Algorithms

If a single illuminant illuminates the scene uniformly, that is, $L_i(x, y) = L_i$, then it suffices to estimate a three component vector $\tilde{\mathbf{L}}$ from all the measured sensor responses with $\tilde{\mathbf{L}} \approx \mathbf{L}$. Given an estimate $\tilde{\mathbf{L}}(x, y)$, a color constant descriptor can be computed by dividing the sensor response by the estimate of the illuminant.

$$\begin{aligned} \frac{c_i(x, y)}{\tilde{L}_i(x, y)} &= \frac{G(x, y) R_i(x, y) L_i(x, y)}{\tilde{L}_i(x, y)} \\ &\approx \frac{G(x, y) R_i(x, y) L_i(x, y)}{L_i(x, y)} \\ &= G(x, y) R_i(x, y) \end{aligned} \quad (1)$$

Such an output image will be a shaded reflectance image. In other words, a diagonal color transform suffices if the sensor response is very narrow band. Some color constancy algorithms, however, also use a general 3×3 matrix transform to compute a color-corrected output image.

It is also possible to transform a given image taken under one illuminant \mathbf{L} to another image taken under a different illuminant \mathbf{L}' . This can be done by multiplying each sensor response vector by a diagonal matrix whose elements are set to $k_i = \frac{L'_i}{L_i}$. The coefficients k_i are called von Kries coefficients [6]. Necessary and sufficient conditions on whether von Kries chromatic adaptation gives color constancy have been derived by West and Brill [7].

A simple algorithm to estimate the color of the illuminant is the white patch Retinex algorithm. It is a simplified version of the parallel Retinex algorithm [8]. In order to understand how this algorithm works, suppose that a white patch, that is, a uniformly reflecting patch, is contained in the imaged scene which is uniformly illuminated. Assuming a normalized sensor response, then the response of the sensors on the white patch will be an estimate of the illuminant. For the white patch, it holds that $R_i = 1$ which leads to $c_i(\text{white patch}) = GL_i$. The white patch algorithm treats each color band separately and searches for the maximum response which is assumed to be an estimate of the illuminant.

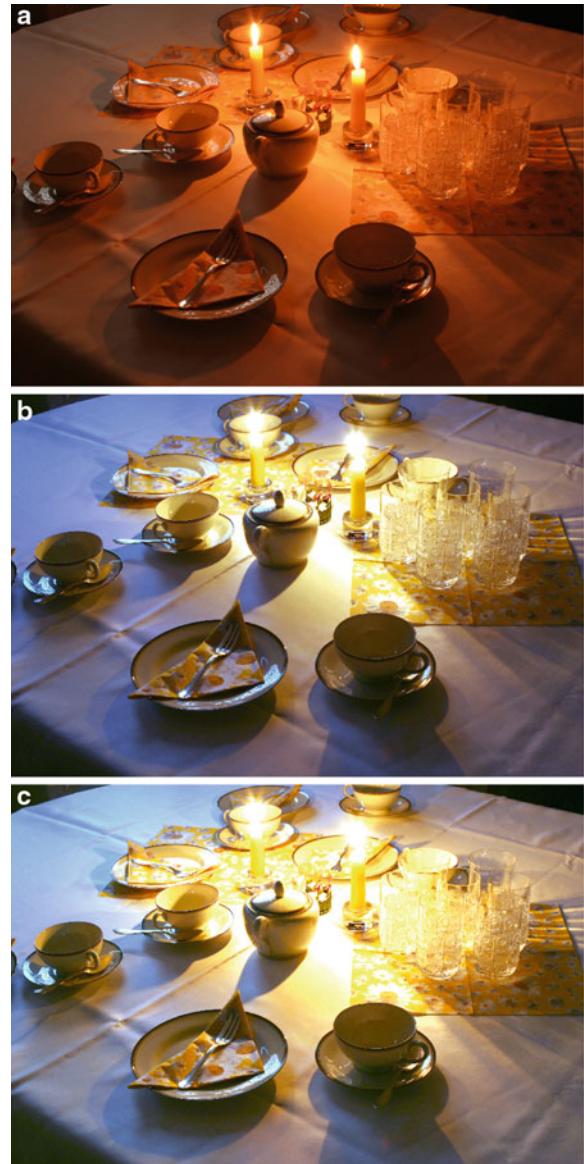
$$\tilde{L}_i = \max_{x,y} c_i(x, y)$$

Instead of locating the maximum response, one can also compute a histogram of the sensor responses and then set the estimate of the illuminant at some percentage from above. This will lead to a more robust estimate of the illuminant. Fig. 1b shows the output of the white patch Retinex algorithm for a sample image shown in Fig. 1a. The illuminant was estimated at 5% from above using a histogram approach for each color band.

Another simple algorithm is based on the gray world assumption which is due to Buchsbaum [9]. According to Buchsbaum, the world is gray on average. Let a_i be the global average of all sensor responses for color channel i where n is the number of sensors.

$$a_i = \frac{1}{n} \sum_{x,y} c_i(x, y) = \frac{1}{n} \sum_{x,y} G(x, y) R_i(x, y) L_i(x, y)$$

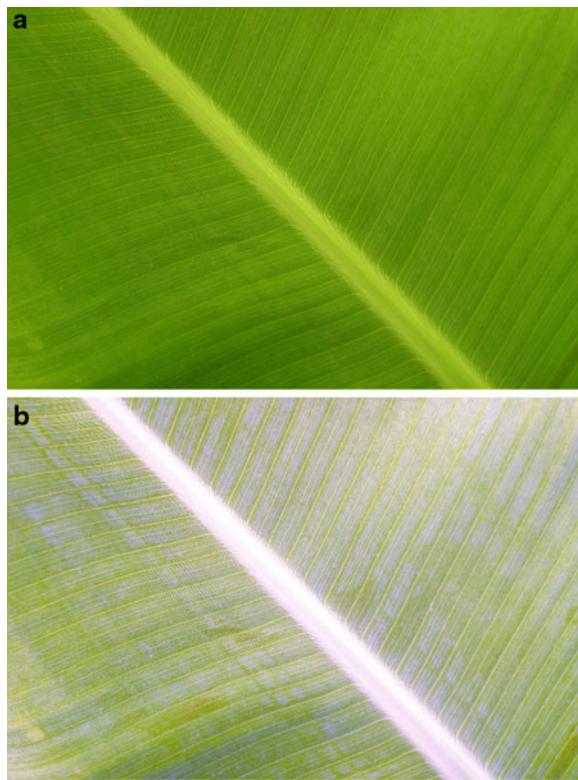
Assuming a uniform illuminant $L_i(x, y) = L_i$ and an independence between shading and reflectance, then:



Color Constancy, Fig. 1 (a) sample input image (b) results for the white patch Retinex algorithm using a histogram (c) results for the gray world assumption

$$\begin{aligned} a_i &= L_i \frac{1}{n} \sum_{x,y} G(x, y) R_i(x, y) \\ &= L_i E[G(x, y) R_i(x, y)] = L_i E[G(x, y)][R_i(x, y)] \end{aligned} \quad (2)$$

where $E[x]$ denotes the expected value of x . Suppose that a large number of differently colored objects are contained in the scene. Thus, a uniform distribution



Color Constancy, Fig. 2 (a) image of a leaf from a banana plant (b) results for the gray world assumption

of colors is assumed. This results in $E[R_i(x, y)] = \frac{1}{n} \sum_{x,y} R_i(x, y) = \frac{1}{2}$ assuming a range of $[0, 1]$ for reflectances and $E[G(x, y)] = c$ where c is a constant. The result is:

$$a_i = \frac{c}{2} L_i.$$

Hence, the illuminant is proportional to global space average color $L_i \propto a_i$ and an estimate of the illuminant can be obtained by setting:

$$\tilde{L}_i = 2a_i.$$

Using $c = 1$ assumes that all patches are frontally oriented or alternatively, that the geometry factor is subsumed into a combined reflectance and geometry factor. [Fig. 1c](#) shows the results for the gray world assumption on a sample image. [Fig. 2](#) shows the results for another image where the assumption, that a large number of different colored objects are contained in the scene, is not fulfilled. In this case, the gray world assumption will not work correctly.

Additional Color Constancy Algorithms

Other important algorithms include gamut constraint algorithms originally developed by Forsyth [10]. A gamut constraint algorithm looks at the gamut of colors contained in a sample image and then transforms this color gamut to a color gamut of a canonical image. Forsyth's algorithm operates on three color channels. Finlayson [11] has developed a variant which operates with a projected color gamut (2D gamut constraint algorithm). Van de Weijer et al. [12] have introduced the gray-edge hypothesis. While the gray-world assumption suggests that the world is gray on average, the gray-edge hypothesis suggests that image derivatives are gray on average. Brainard and Freeman [13] have addressed the problem of color constancy using Bayesian decision theory.

Uniform Versus Nonuniform Illumination

Many color constancy algorithms assume that the scene is uniformly illuminated by a single illuminant. If multiple light sources are distributed over the scene, then it is assumed that these light sources can be combined into a single average illuminant. This is possible provided that the light sources are sufficiently distant from the scene. If the illumination is uniform, then only a three component descriptor has to be estimated from the input image. However, in practice, many scenes are illuminated nonuniformly. Very often, one has several different illuminants. For instance, daylight may be falling through a window while artificial lights have already been turned on inside a building. Nonuniform illumination may also be present outside during a sunny day. Consider a family sitting in the garden under a red umbrella and a photographer taking a photograph of the family members. The family would be illuminated by light reflected from the red umbrella while the surrounding would be illuminated by direct sunlight. A digital camera usually corrects for a single illuminant. Thus, either the family members would have a red color cast to them or the background colors would not look right in the resulting image.

Algorithms have also been developed which can cope with a spatially varying illuminant. Land and McCann's Retinex algorithm [8] is a parallel algorithm for color constancy which allows for a nonuniform illumination. They only considered one-dimensional paths over the sensor array. Horn [14] extended Land's Retinex algorithm to two dimensions.

Blake [15] provided additional improvements. Barnard et al. [16] extended the 2D gamut constraint algorithm to scenes with varying illumination.

Ebner [1] showed how a grid of processing elements is able to estimate the color of the illuminant locally using the gray world assumption. Each processing element receives the measurement c_i from the sensor and computes local space average color. The processing elements are laterally connected to neighboring processing elements. Let $a_i(x, y)$ be the current estimate of local space average color for a processing element located at position (x, y) . Each processing element receives the estimate of local space average color from neighboring elements and averages the neighboring estimates to update its own estimate. A small component from the sensor measurement is then added to this estimate. Let p be a small percentage and let $N(x, y)$ be the neighborhood of the processing element at position (x, y) , then the computation of a processing element consists of the following two updates.

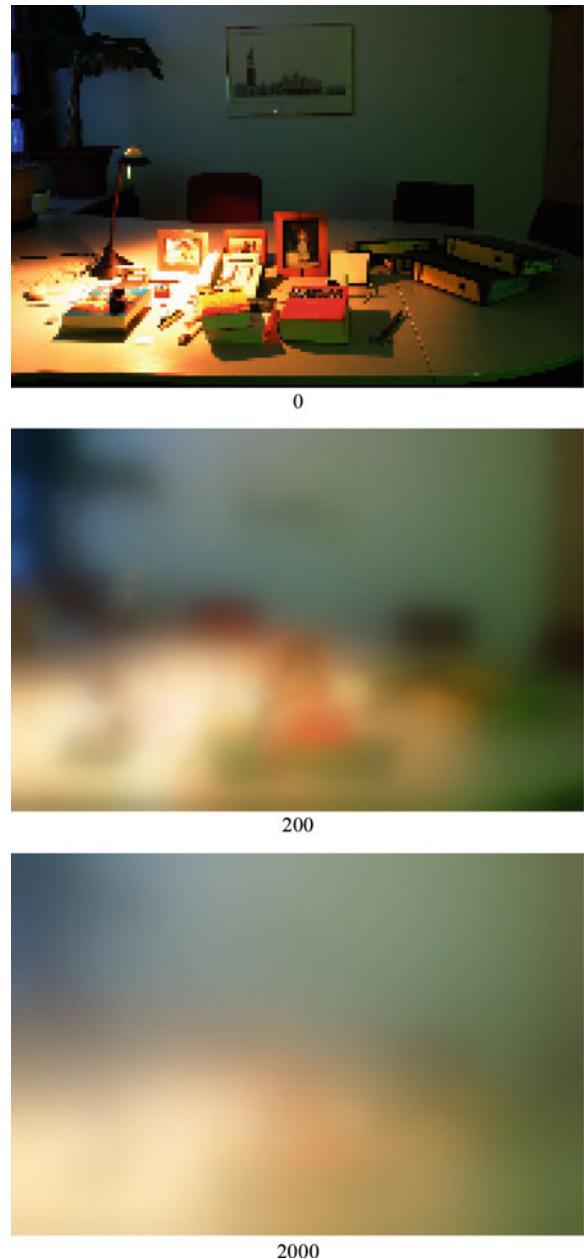
$$a_i'(x, y) = \frac{1}{|N(x, y)|} \sum_{(x', y') \in N(x, y)} a_i(x', y')$$

$$a_i(x, y) = (1 - p)a_i'(x, y) + pc_i(x, y)$$

The two updates are carried out iteratively. This process converges to local space average color which is an estimate of the illuminant, $\tilde{L}_i(x, y) = 2a_i(x, y)$. Figure 3 shows local space average color after 0, 200, and 2,000 iterations using $p = 0.0001$ given the input image shown in Fig. 4b. The extent of the averaging is determined by the parameter p . Figure 4 shows the output when local space average color is used to estimate the illuminant locally. Local average color was computed for a downsampled image (25% in each direction), the original image had 768×512 pixels. Ebner [1] suggested that a similar algorithm may be used by the brain for color perception.

Advanced Reflectance Models

The theoretical model of color image formation, that has been given above, assumed that objects diffusely reflect the incident light. This is not the case for all surfaces, for example, brushed metal or plastics. Especially for plastic objects or objects covered with gloss varnish, a more elaborate reflectance model is



Color Constancy, Fig. 3 Local space average color after 0, 200 and 2,000 iterations

more appropriate. The dichromatic reflection model [17, 18] assumes that reflectance is composed of interface reflectance, which occurs at the boundary between the object's surface and air, and body reflectance which is due to the scattering of light below the object's surface. In other words, the reflection of light from the object's surface is assumed to be partially specular



Color Constancy, Fig. 4 (a and b) input images (c and d) results using local space average color

and partially diffuse. Color constancy algorithms have also been developed using the dichromatic reflection model, for example, by Risson [19].

Application

Color constancy algorithms are ideal for color correction in digital photography. In digital photography, the goal is to obtain a color-corrected image that corresponds nicely to human perception. A printed photograph or a photograph viewed on a computer display should appear in exactly the same way that the human observer (the photographer) perceived the scene. Besides digital photography, color constancy algorithms can be applied in the context of most computer vision tasks. For many tasks, one should try to estimate object reflectances. For instance, image segmentation would not be as difficult, if object reflectance could be correctly determined. Similarly, color-based object recognition is easier if performed on reflectance information. Thus, color constancy

algorithms should often be applied as a preprocessing step. This holds especially for autonomous mobile robots equipped with a vision system because autonomous mobile robots need to operate in different environments under different illuminants.

Experimental Results and Datasets

A comparison of computational color constancy algorithms is given by Barnard et al. [20] for synthetic as well as real image data. Another detailed comparison of color constancy algorithms along with pseudocode for many algorithms is given by Ebner [1]. Data for computer vision and computational color science can be found at the Simon Fraser University, Canada (www.cs.sfu.ca/~colour/data/). A repository has also been created by the color group at the University of East Anglia, UK (www.colour-research.com). A database for spectral color science is available from the University of Eastern Finland, Finland (spectral.joensuu.fi/).

References

1. Ebner M (2007) Color constancy. Wiley, England
2. Maloney LT (1999) Physics-based approaches to modeling surface color perception. In: Gegenfurtner KR, Sharpe LT (eds) *Color vision: from genes to perception*. Cambridge University Press, Cambridge, pp 387–422
3. Zeki S (1993) *A vision of the brain*. Blackwell, Oxford
4. Geusebroek JM, van den Boomgaard R, Smeulders AWM, Geerts H (2001) Color invariance. *IEEE Trans Pattern Anal Mach Intell* 23(12):1338–1350
5. Finlayson GD, Hordley SD (2001) Color constancy at a pixel. *J Opt Soc Am A* 18(2):253–264
6. Richards W, Parks EA (1971) Model for color conversion. *J Opt Soc Am* 61(7):971–976
7. West G, Brill MH (1982) Necessary and sufficient conditions for von Kries chromatic adaptation to give color constancy. *J Math Biol* 15:249–258
8. Land EH, McCann JJ (1971) Lightness and retinex theory. *J Opt Soc Am* 61(1):1–11
9. Buchsbaum G (1980) A spatial processor model for object colour perception. *J Frankl Inst* 310(1):337–350
10. Forsyth DA (1990) A novel algorithm for color constancy. *Int J Comput Vis* 5(1):5–36
11. Finlayson GD (1996) Color in perspective. *IEEE Trans Pattern Anal Mach Intell* 18(10):1034–1038
12. van de Weijer J, Gevers T, Gijsenij A (2007) Edge-based color constancy. *IEEE Trans Image Process* 16(9): 2207–2214
13. Brainard DH, Freeman WT (1997) Bayesian color constancy. *J Opt Soc Am A* 14(7):1393–1411
14. Horn BKP (1974) Determining lightness from an image. *Comput Graph Image Process* 3:277–299
15. Blake A (1985) Boundary conditions for lightness computation in mondrian world. *Comput Vis Graph Image Process* 32:314–327
16. Barnard K, Finlayson G, Funt B (1997) Color constancy for scenes with varying illumination. *Comput Vis Image Underst* 65(2):311–321
17. Klinker GJ, Shaffer SA, Kanade T (1988) The measurement of highlights in color images. *Int J Comput Vis* 2:7–32
18. Tominaga S (1991) Surface identification using the dichromatic reflection model. *IEEE Trans Pattern Anal Mach Intell* 13(7):658–670
19. Risson VJ (2003) Determination of an illuminant of digital color image by segmentation and filtering. United States Patent Application, Pub. No. US 2003/0095704 A1
20. Barnard K, Cardei V, Funt B (2002) A comparison of computational color constancy algorithms – Part I and II. *IEEE Trans Image Process* 11(9):972–996

Color Difference

- [Color Similarity](#)

Color Discrepancy

- [Color Similarity](#)

Color Dissimilarity

- [Color Similarity](#)

Color Management

- [Gamut Mapping](#)

Color Model

Shoji Tominaga
Graduate School of Advanced Integration Science,
Chiba University, Inage-ku, Chiba, Japan

Synonyms

[Color spaces](#); [Color specification systems](#)

Definition

A color model is a mathematical model describing the way colors are specified by three-dimensional coordinates in such terms as three numerical values, three perceptual attributes, or three primary colors.

Background

The human retina contains three types of cones L, M, and S as photoreceptors, which respond best to light of long, medium, and short wavelengths, respectively, in the visible range. Since the perception of

color depends on the response of these cones, it follows in principle that visible color can be mapped into a three-dimensional space in terms of three numbers.

Trichromatic color vision theory is closely related to such three types of color sensors. The trichromatic theory by Young and Helmholtz suggests that any colors and spectra in the visible range can be visually matched using mixtures of three primary colors. These primary colors are usually red, green, and blue. Most image output devices such as television screens, computer and video displays, and image projectors create visible colors by using additive mixtures of the three primary colors.

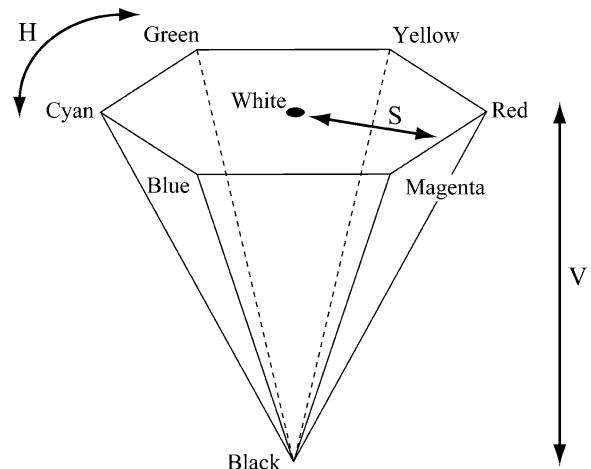
Digital cameras also capture color by using the same principle of trichromatic color vision. They usually use three different types of sensors which primarily respond to the red, green, and blue parts of the visible spectrum. Therefore, a color model mathematically describing the color coordinate system is crucial for analysis, evaluation, and the rendering of color images.

Theory

Color Model for RGB Color Signals

Most color models used in computer vision and image processing are based on the RGB primaries. This is because color images captured by digital cameras are all represented by the RGB primaries, and image output devices such as displays and projectors are based on an additive mixture of the RGB primaries. Therefore, the simplest color model (called RGB color space) uses a Cartesian coordinate system defined by (R, G, B) triples. This system, however, is not available for colorimetry, because the spectral-sensitivity curves of cameras are generally not coincident with the color-matching functions.

The RGB space is not a perceptually uniform color space. In computer graphics and image processing applications, approximately uniform spaces derived from RGB are defined in terms of the three attributes: hue (H), saturation (S), and value (V) (representing lightness). For example, the HSV model by Smith [9] was defined as a nonlinear transform of RGB given in 8 bits:



Color Model, Fig. 1 Single hexagonal color space

$$\begin{aligned} H &= \tan^{-1} \left\{ \frac{\sqrt{3}(g-b)}{(r-g)+(r-b)} \right\} \\ S &= 1 - \text{Min}/V \\ V &= \text{Max} \end{aligned} \quad (1)$$

where

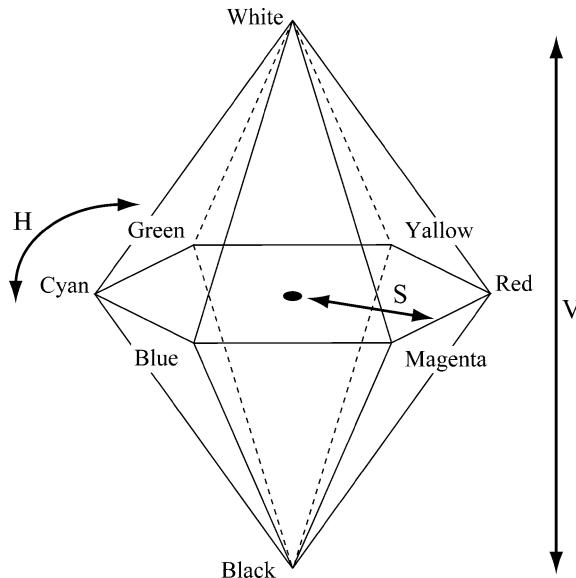
$$\begin{aligned} r &= R/255, \quad g = G/255, \quad b = B/255, \\ \text{Max} &= \max(r, g, b), \quad \text{Min} = \min(r, g, b). \end{aligned}$$

This model represents a hexagonal color space as seen in Fig. 1, where the saturation decreases monotonically as the value decreases. Figure 2 represents the improved HSV model to a double hexagonal space, which is defined as:

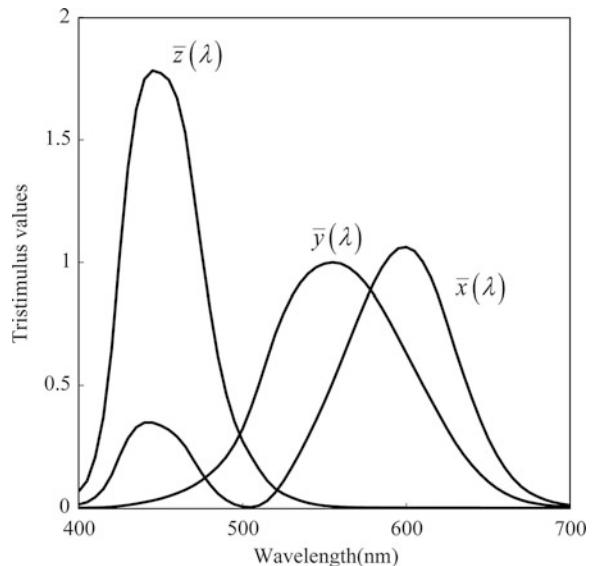
$$\begin{aligned} H &= \tan^{-1} \left\{ \frac{\sqrt{3}(g-b)}{(r-g)+(r-b)} \right\} \\ S &= \text{Max} - \text{Min} \\ V &= (R+g+b)/3 \end{aligned} \quad (2)$$

Color Model for Colorimetry

The CIE-XYZ color system for colorimetry was created by the International Commission on Illumination (CIE). This color system was derived from a series of visual experiments of color matching [1]. The tristimulus values of a color were used for representing



Color Model, Fig. 2 Double hexagonal color space



Color Model, Fig. 3 CIE 1931 2° Standard Observer color-matching functions

the amounts of three primary colors needed to match the test color. The tristimulus values depend on the observer's field of view. Therefore, the CIE defined the standard observer and a set of three color-matching functions, called $\bar{x}(\lambda)$, $\bar{y}(\lambda)$, and $\bar{z}(\lambda)$. The chromatic response of the standard observer is characterized by the three color-matching functions. Figure 3 shows the CIE 1931 2° standard observer color-matching functions.

The tristimulus values for a color signal with a spectral-power distribution $L(\lambda)$ are then given by:

$$\begin{aligned} X &= k \int_{\lambda} L(\lambda) \bar{x}(\lambda) d\lambda \\ Y &= k \int_{\lambda} L(\lambda) \bar{y}(\lambda) d\lambda \\ Z &= k \int_{\lambda} L(\lambda) \bar{z}(\lambda) d\lambda \end{aligned} \quad (3)$$

where the integration on the wavelength λ is normally calculated over the range of visible spectrum 400–700 nm, and the coefficient k is a normalizing constant. In application to photometry, the Y tristimulus value becomes the luminance of the color signal. In application to object color, the tristimulus values become:

$$\begin{aligned} X &= k \int_{\lambda} E(\lambda) S(\lambda) \bar{x}(\lambda) d\lambda \\ Y &= k \int_{\lambda} E(\lambda) S(\lambda) \bar{y}(\lambda) d\lambda \\ Z &= k \int_{\lambda} E(\lambda) S(\lambda) \bar{z}(\lambda) d\lambda \end{aligned} \quad (4)$$

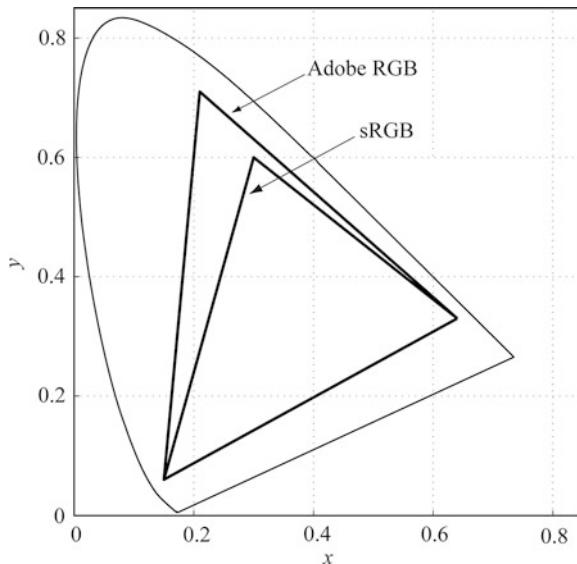
where $S(\lambda)$ is the surface-spectral reflectance of the illuminated object, and $E(\lambda)$ is the spectral-power distribution of the light source illuminating the object. In this case, the Y tristimulus value becomes the luminance factor of the object-color stimulus. The constant factor is usually chosen as:

$$k = 100 \int_{\lambda} S(\lambda) \bar{x}(\lambda) d\lambda. \quad (5)$$

The chromaticity coordinates (x, y) of a given color are defined as:

$$x = \frac{X}{X + Y + Z}, \quad y = \frac{Y}{X + Y + Z}. \quad (6)$$

Figure 4 depicts the chromaticity diagram, where all visible chromaticities are located within the horseshoe-shaped region. The outer curved boundary is the spectral locus, corresponding to monochrome light with the



Color Model, Fig. 4 CIE 1931 chromaticity diagram

most saturated color. Less saturated colors are located in the inside of the region with white at the center. The triangle of sRGB represents the gamut of a standard RGB color space proposed by HP and Microsoft in 1996 [10]. Recently, usual monitors, printers, and the Internet are based on this standard space. The triangle of Adobe RGB represents the gamut of an RGB color space developed by Adobe Systems in 1998 [11]. This color space, improving the gamut of sRGB primarily in cyan-green, is used for desktop publishing.

It should be noted that the CIE XYZ tristimulus values were not defined for color differences. Two colors with a constant difference in the tristimulus values may look very different, depending on where the two colors are located in the (x, y) chromaticity diagram.

Color Model for Uniform Color Space

Many people believe that a uniform color space is most useful for applications where perceptual errors are evaluated. The CIE 1976 L*a*b* color system (also called CIE LAB color system) was designed to approximate a perceptually uniform color space in terms of the tristimulus values XYZ. It is defined as follows:

$$L^* = 116(Y/Y_n)^{1/3} - 16$$

$$a^* = 500[(X/X_n)^{1/3} - (Y/Y_n)^{1/3}] \quad (7)$$

$$b^* = 200[(Y/Y_n)^{1/3} - (Z/Z_n)^{1/3}]$$

where (X_n, Y_n, Z_n) are the tristimulus values of the reference white point. This system provides an opponent-color space with dimension L^* for lightness, and a^* and b^* for opponent-color dimensions. $L^*=0$ and $L^*=100$ indicate black and white, respectively. The axes of a^* and b^* take the position between red and green, and the position between yellow and blue, respectively.

The two-dimensional chromaticity diagram is expressed in the orthogonal coordinate system of (a^*, b^*) . The chromaticity components can also be expressed in a cylindrical system of chroma and hue, where the chroma and hue-angle are defined respectively by:

$$C_{ab}^* = (a^{*2} + b^{*2})^{1/2} \quad (8)$$

$$h_{ab} = \arctan^{-1}(b^*/a^*)$$

The color difference between two color stimuli 1 and 2 is defined as follows:

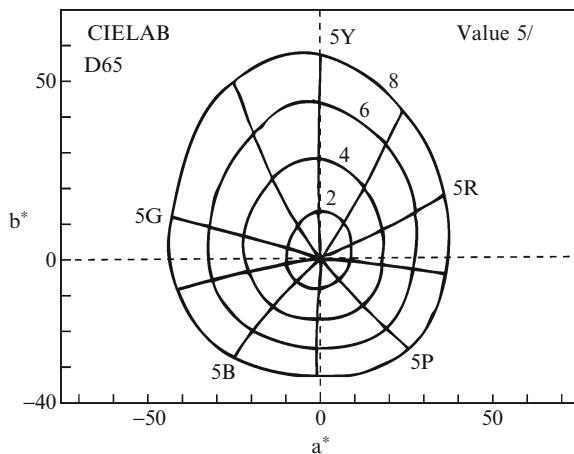
$$\Delta E_{ab}^* = [(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2]^{1/2}, \quad (9)$$

where $\Delta L^* = L_1^* - L_2^*$, $\Delta a^* = a_1^* - a_2^*$, and $\Delta b^* = b_1^* - b_2^*$. Figure 5 depicts the Munsell color system with value 5 on the (a^*, b^*) chromaticity coordinates under the reference condition of illuminant D_{65} . The color system approximates roughly the Munsell uniform color space.

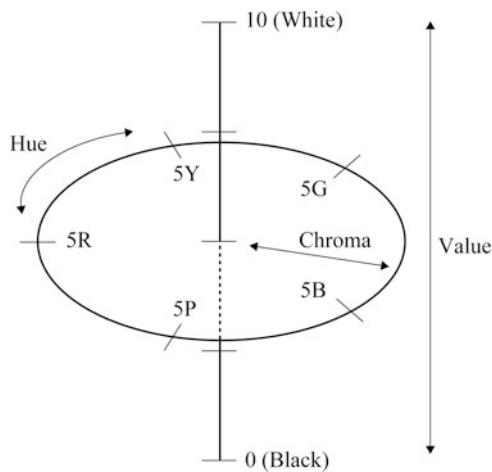
Color Model for Color Order System

Color order systems were created to order colors in terms of intuitive perceptual attributes. Most colored samples, used as a reference in many design and engineering applications, are made in equal steps of perceptual attributes.

The Munsell color order system is one of the most widely used systems in the world. This system is based on the three perceptual attributes: hue, value (representing lightness), and chroma (representing saturation), and is organized in a cylindrical coordinate system as shown in Fig. 6. Hue is arranged as a circle,



Color Model, Fig. 5 Munsell color system with Value 5 on the (a^*, b^*) chromaticity coordinates



Color Model, Fig. 6 Organization of the Munsell color system

value as the axis perpendicular to the hue circle, and chroma as the distance from the center.

The Munsell color samples are arranged in equal steps of each attribute, where each sample is described using a three-part syntax of hue chroma/value. For example, 5YR 8/4 indicates a color sample with the hue of 5YR (yellow-red), the chroma of level 4, and the value of level 8. A conversion table to reproduce the Munsell color samples in terms of the CIE tristimulus values was given by Newhall et al. [12]. The table contains the luminance factor Y and chromaticity coordinates (x, y) equivalent to the Munsell renotation system (H, V, C), under the condition of the CIE standard illuminant C. It should be noted that there is no

formula for a color notation conversion between the Munsell and CIE system, except for the table. Therefore, the conversion was carried out by interpolating the table data.

Obviously, this method is not efficient from the point of view of mass data processing by computers. Tominaga [13] developed a method for color notation conversion, between the Munsell and CIE systems by means of neural networks. In the neural network method, it is not necessary to use the special database of color samples. This is because the knowledge of conversion between the two color spaces is stored in a small set of the weighting parameters in a multilayer feedforward network.

References

- Wyszecki G, Stiles WS (1982) Color science: concepts and methods, quantitative data and formulae. Wiley, New York
- Hall R (1989) Illumination and color in computer-generated imagery. Springer, New York
- Foley JD, van Dam A, Feiner SK, Hughes JF (1990) Computer graphics: principles and practice, 2nd edn. Addison-Wesley, Reading
- Wandell BA (1995) Foundations of vision. Sinauer Associates, Sunderland
- Shevell SK (2003) The science of color, 2nd edn. Elsevier, Oxford
- Lee H-C (2005) Introduction to color imaging science. Cambridge University Press, Cambridge
- Fairchild MD (2005) Color appearance models. Wiley, Chichester
- Shanda J (2007) Colorimetry. Wiley, Hoboken
- Smith AR (1978) Color gamut transform pairs. Comput Graph 12(3):12–19
- Stokes M, Anderson M, Chandrasekar S, Motta R (1996) A standard default color space for the Internet: sRGB. Version 1.10, International Color Consortium
- Adobe Systems (2005) Adobe RGB(1998)b color image encoding. Version 2005-5, Adobe Systems
- Newhall SM, Nickerson D, Judd DB (1943) Final report of the O.S.A. subcommittee on the spacing of the munsell colors. J Opt Soc Am 33(7):385–411
- Tominaga S (1992) Color classification of natural color images. Color Res Appl 17(4):230–239
- Brainard DH (1989) Calibration of a computer controlled color monitor. Color Res Appl 14:23–34
- Pratt WK (1991) Digital image processing, 2nd edn. Wiley, New York
- Ohta N, Robertson AR (2005) Colorimetry. Wiley, Chichester
- Hunt RWG (2004) The reproduction of colour, 6th edn. Wiley, Chichester
- Green P (2010) Color management. Wiley, Chichester
- Watt A (2000) 3D computer graphics, 3rd edn. Addison Wesley, Reading



Color Similarity

Takahiko Horiuchi and Shoji Tominaga
Graduate School of Advanced Integration Science,
Chiba University, Inage-ku, Chiba, Japan

Synonyms

Color difference; Color discrepancy; Color dissimilarity

Definition

Color similarity is a measure that reflects the strength of relationship between two colors.

Background

A measure of similarity between colors is needed when one color region is matched to another region among color images, for example, a purple region is more similar to a blue region than a green region. Therefore, the similarity between two colors is an important metric in not only color science but also imaging science and technology (including computer vision). If color similarity is measured, it allows people to determine the strength of relationship in terms of numerical values; otherwise, it will be determined in nonnumerical ways.

Thus, the benefit of color similarity is its application to various techniques of image analysis, such as noise removal filters, edge detectors, object classification, and image retrieval.

Theory

Color similarity is usually represented in the range of either interval $[-1, 1]$ or interval $[0, 1]$. Let $\mathbf{x}_i = [x_{i1}, x_{i2}, x_{i3}]$ be a three-dimensional vector specifying a color feature at location i in a color image. The similarity measure $s(\mathbf{x}_i, \mathbf{x}_j)$ between two color features is a symmetric function whose value is larger when color features \mathbf{x}_i and \mathbf{x}_j are closer. The color feature is specified by three coordinates in color space (or color model). Color space is defined in many different ways

in color and image science, where color spaces such as RGB, $L^*a^*b^*$, $L^*u^*v^*$, YIQ, and HSI are often used.

On the other hand, color dissimilarity is also used for measuring the discrepancy between two colors. This quantity is usually represented in a positive range of $[0, 1]$ because the dissimilarity is measured by a “normalized distance” between two colors. A dissimilarity measure $d(\mathbf{x}_i, \mathbf{x}_j)$ is a symmetric function whose value is larger when color features \mathbf{x}_i and \mathbf{x}_j are more dissimilar. Therefore, a relationship between two measures of color similarity and color dissimilarity is given by:

$$s(\mathbf{x}_i, \mathbf{x}_j) = 1 - d(\mathbf{x}_i, \mathbf{x}_j), \quad (1)$$

where the color similarity $s(\mathbf{x}_i, \mathbf{x}_j)$ is bounded by $[0, 1]$. Note that when the color similarity is one (i.e., exactly same), the color dissimilarity is zero, and when color similarity is zero (i.e., quite different), the color dissimilarity is one. If the color similarity is defined in the range $[-1, 1]$, then:

$$s(\mathbf{x}_i, \mathbf{x}_j) = 1 - 2d(\mathbf{x}_i, \mathbf{x}_j). \quad (2)$$

Note in this case that the color dissimilarity of 1 (i.e., quite different) corresponds to the color similarity of -1 , and when the color dissimilarity of 0 (i.e., exactly same) corresponds to the color similarity of 1. In many cases, measuring dissimilarity is easier than measuring similarity. Once dissimilarity is measured, it can easily be normalized and converted to the similarity measure. Therefore, color dissimilarity measures are first described in the following.

Color Dissimilarity Measures

The most commonly used dissimilarity measure to quantify the distance between two color vectors \mathbf{x}_i and \mathbf{x}_j is the weighted Minkowski metric:

$$d(\mathbf{x}_i, \mathbf{x}_j) = c \left(\sum_{k=1}^3 \xi_k |x_{ik} - x_{jk}|^L \right)^{1/L}, \quad (3)$$

where c is the nonnegative scaling parameter for $d(\mathbf{x}_i, \mathbf{x}_j) \in [0, 1]$ and the exponent L defines the nature of the distance metric. The parameter ξ_k , for $\sum_k \xi_k = 1$, measures the weight assigned to the color channel k . Usually it is determined $\xi_k = 1/3, \forall k$ (e.g., [1, 2]). In the case $L = 1$ in (Eq. 3), the

measure is known as Manhattan distance, which is also called city-block distance, absolute value distance, and taxicab distance [3].

$$d(\mathbf{x}_i, \mathbf{x}_j) = c \sum_{k=1}^3 |x_{ik} - x_{jk}|. \quad (4)$$

The dissimilarity measure represents distance between points in a city road grid. It examines the absolute differences between a pair of colors. In the case $L = 2$ in (Eq. 3), the measure is known as Euclidean distance which is the most common use of distance.

$$d(\mathbf{x}_i, \mathbf{x}_j) = c \left(\sum_{k=1}^3 (x_{ik} - x_{jk})^2 \right)^{1/2}. \quad (5)$$

In the case $L \rightarrow \infty$ in (Eq. 3), the measure is known as Chebyshev distance which is also called Kolmogorov-Smirnov Statistic [4], chess-board distance, and maximum value distance. It examines the absolute magnitude of the differences between coordinates of a pair of objects.

$$d(\mathbf{x}_i, \mathbf{x}_j) = c \max_k |x_{ik} - x_{jk}|. \quad (6)$$

As a well-known dissimilarity metric based on the Minkowski metric, [6] color difference ΔE_{ab}^* is useful [5, 7]. ΔE_{ab}^* is the Euclidean distance between two colors in CIEL*a*b* color space as follows:

$$\begin{aligned} \Delta E_{ab}^*(\mathbf{x}_i, \mathbf{x}_j) &= \left((L_i^* - L_j^*)^2 + (a_i^* - a_j^*)^2 \right. \\ &\quad \left. + (b_i^* - b_j^*)^2 \right)^{1/2}, \end{aligned} \quad (7)$$

where $\mathbf{x}_i = [L_i^*, a_i^*, b_i^*]$ and $\mathbf{x}_j = [L_j^*, a_j^*, b_j^*]$. Please note that ΔE_{ab}^* is used as a “distance” with $c = 1$ in (Eq. 5). When $\Delta E_{ab}^* \approx 2.3$, it corresponds to a just noticeable difference (JND) of surface colors [8, 9]. Perceptual nonuniformities in the underlying CIEL*a*b* color space, however, have led to the CIE refining their 1976 definition over the years. The refined color difference ΔE_{94}^* is defined in the L*C*h color space [10]. Given a reference color $\mathbf{x}_i = [L_i^*, C_i^*, h_i^*]$ and another color $\mathbf{x}_j = [L_j^*, C_j^*, h_j^*]$,

the difference is:

$$\begin{aligned} \Delta E_{94}^*(\mathbf{x}_i, \mathbf{x}_j) &= \left(\left(\frac{L_i^* - L_j^*}{K_L} \right)^2 + \left(\frac{C_i^* - C_j^*}{1 + K_1 C^*} \right)^2 \right. \\ &\quad \left. + \left(\frac{h_i^* - h_j^*}{1 + K_2 C^*} \right)^2 \right)^{1/2} \end{aligned} \quad (8)$$

where C^* represents the geometrical average of chroma, and K_L , K_1 , and K_2 represent the weighting factors which depend on the application (in original definition, $K_L = 1$, $K_1 = 0.045$, and $K_2 = 0.015$ for graphic arts). Since the 1994 definition did not adequately resolve the perceptual uniformity issue, the CIE refined their definition, adding five corrections:

1. A hue rotation term (RT), to deal with the problematic blue region
2. Compensation for neutral colors
3. Compensation for lightness
4. Compensation for chroma
5. Compensation for hue

The definition of the CIEDE2000 color difference is explained in Ref. [11].

Color Similarity Measure

The simplest color similarity measure is an angular separation similarity. It represents a cosine angle between two color vectors, which is often called as coefficient of correlation. In this measure, similarity in orientation is expressed through the normalized inner product as:

$$s(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j / |\mathbf{x}_i| |\mathbf{x}_j| \quad (9)$$

which corresponds to the cosine of the angle between the two color vectors \mathbf{x}_i and \mathbf{x}_j . Since similar colors have almost parallel orientations, (*while significantly different colors point in different directions in a 3-D color space*) the normalized inner product can be used to quantify orientation similarity between the two color vectors.

The correlation coefficient is a standardized angular separation by centering the coordinates to their mean value. Let $\bar{x}_i = \frac{1}{p} \sum_{k=1}^p x_{ik}$ and $\bar{x}_j = \frac{1}{p} \sum_{k=1}^p x_{jk}$; the correlation coefficient between these two vectors \mathbf{x}_i and \mathbf{x}_j is given as follows:



$$s(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{k=1}^p |x_{ik} - \bar{x}_i| |x_{jk} - \bar{x}_j|}{\left(\sum_{k=1}^p (x_{ik} - \bar{x}_i)^2\right)^{1/2} \left(\sum_{k=1}^p (x_{jk} - \bar{x}_j)^2\right)^{1/2}}. \quad (10)$$

In (Eq. 10) the color similarity between two points is shown as $p = 3$ shows. Then the correlation coefficient can be applied to color region similarity, such as textured regions by setting $p = 3 \times n$ (where n is the number of pixels).

There are many other methods to measure the similarity between two color vectors. So depending on the nature and the objective of the problem at hand, it is possible that one method is more appropriate than the other.

Application

The formulation of similarity between two color vectors is of paramount importance for the development of the vector processing techniques. These include noise removal filters, edge detectors, image zoomers, and image retrievals.

References

1. Kruskal JB (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika* 29(1):1–27. doi:10.1007/BF02289565
2. Duda RO, Hart PE (1973) Pattern classification and scene analysis. Wiley, New York
3. Krause EF (1987) Taxicab geometry: adventure in non-euclidean geometry. Dover, New York. ISBN 0-486-25202-7
4. Geman D, Geman S, Graffigne C, Dong P (1990) Boundary detection by constrained optimization. *IEEE Trans Pattern Anal Mach Intell* 12(7):609–628
5. Pauli H (1976) Proposed extension of the CIE recommendation on uniform color spaces, color difference equations, and metric color terms. *J Opt Soc Am* 66(8):866–867. doi:10.1364/JOSA.66.000866
6. CIE (2004) Colorimetry. Vienna, CIE Pb. 15:2004, ISBN 978-3-901906-33-6
7. CIE Publication (1978) Recommendations on uniform color spaces, color difference equations, psychometric color terms. 15(Suppl 2), Paris, (E.-1.3.1) 1971/(TC-1.3)
8. Sharma G (2003) Digital color imaging handbook, 1.7.2 edn. CRC, Florida. ISBN 084930900X
9. Mahy M, Eycken LV, Oosterlinck A (1994) Evaluation of uniform color spaces developed after the adoption of CIELAB and CIELUV. *Color Res Appl* 19(2): 105–121

10. CIE Publication (1995) Industrial colour-difference evaluation. Vienna, CIE 116–1995, ISBN 978-3-900734-60-2
11. CIE Publication (2001) Improvement to industrial colour-difference evaluation. Vienna, CIE 142–2001, ISBN 978-3-901906-08-4

Color Spaces

Rajeev Ramanath¹ and Mark S. Drew²

¹DLP® Products, Texas Instruments Incorporated, Plano, TX, USA

²School of Computing Science, Simon Fraser University, Vancouver, BC, Canada

Synonyms

Color appearance models; Color model

Definition

A color space describes the range of colors – the *gamut* – that an imaging device or software has to work with. Consequently, the design of these color spaces allows a user to modify images in a predefined manner based on the specific needs of the application.

Theory

Color spaces may be generally separated into those that are defined for analysis of color by color scientists (*colorimetric color spaces*) and those that are used for image/color editing.

Color Spaces for Colorimetric Analysis

Color spaces for colorimetric analysis are typically based on the human observer. Central to such color spaces is the CIEXYZ *color matching functions*, which are based on CIERGB color matching functions – based in turn on the LMS cone fundamentals. CIE is an abbreviation for the International Commission on Illumination (Commission Internationale de L'Eclairage) that is the body that is responsible for standardization of data in this field. However, the ability to directly measure the cone functions of an observer is only a recent development. Researchers had originally



inferred the cone functions based on psychophysical experiments measuring the RGB color matching functions. Consequently, the CIE had standardized the color matching functions long before the cone functions could be directly measured. Details of how the color matching functions were developed are well documented in the book by Wyszecki and Stiles that is arguably a cornerstone of colorimetry [27].

The RGB color matching functions, denoted $\{\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)\}$, are shown in Fig. 1a.

Note that these color matching functions have negative excursions, due to the fact that there are some colors that reside outside the triangle formed by these three primaries – a negative excursion in the primaries' weights is the only way to represent a color outside this triangle. In order to address this problem, the CIE also published a set of color matching functions with nonnegative values, denoted $\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda)$ and shown in Fig. 1b. These are known generally as the CIEXYZ color matching functions. Note that these are now fictitious, nonphysical primaries. The equations for computing these special “tristimulus” values, in XYZ color space, are:

$$\begin{aligned} X &= k \int_{\lambda} \bar{x}(\lambda) i_r(\lambda) d\lambda \\ Y &= k \int_{\lambda} \bar{y}(\lambda) i_r(\lambda) d\lambda \\ Z &= k \int_{\lambda} \bar{z}(\lambda) i_r(\lambda) d\lambda. \end{aligned} \quad (1)$$

where $i_r(\lambda)$ denotes the spectral power distribution of the light energy incident on the retina, and k denotes a normalization factor that is set to 683 lumens/Watt in the case of absolute colorimetry and to $100 / \int_{\lambda} \bar{y}(\lambda) i_r(\lambda) d\lambda$ for relative colorimetry. In the case of relative colorimetry, this means a value of $Y = 100$ denotes the brightest color – the illuminant reflecting from a perfect reflecting diffuser.

The first set of color matching functions published by the CIE were originally empirically determined for a 2° field – the bipartite field used for matching a subtended 2° angle on the observers' retina. Following the 1931 publication, W. S. Stiles and J. M. Burch conducted experiments [25] to measure color matching functions for larger fields of view. This was combined with the findings of Speranskaya [24] into the publication by the CIE of a 10° observer in 1964 [5]. The difference between these two standard observers

is significant enough to warrant a clear specification of which observer color matching functions are used in experimental work. More specifically, the 10° observer has noticeable shifts of the color matching functions in the blue direction due to the fact that the subtense of the stimulus encompasses a larger portion of the retina and hence more S cones and also increased macular pigment absorption.

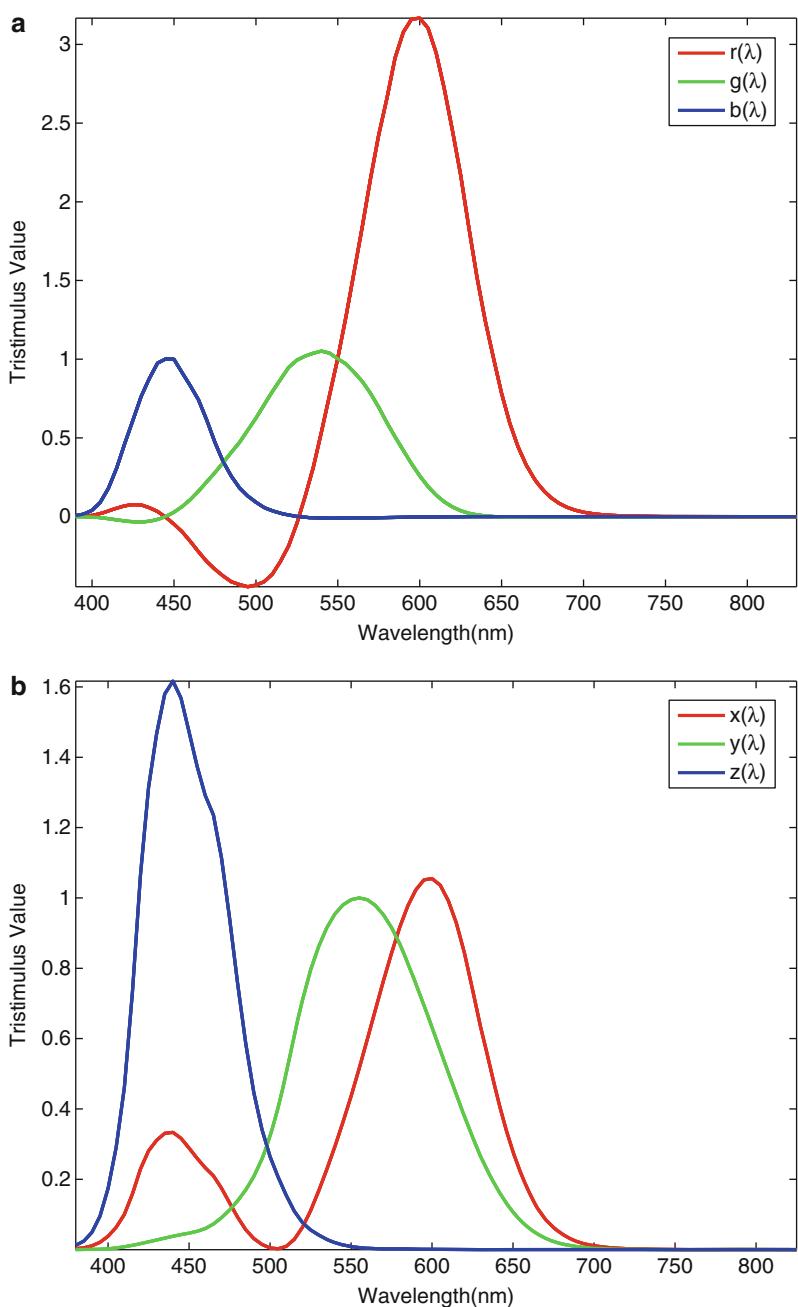
In the CIE colorimetric system, an XYZ tristimulus value uniquely specifies a color. However, a convenient 2D representation of the tristimulus values led to the projection of the tristimulus values by normalizing by the sum of the three values. These “chromaticity” values are given by:

$$\begin{aligned} x &= \frac{X}{X + Y + Z} \\ y &= \frac{Y}{X + Y + Z}. \end{aligned} \quad (2)$$

A color may be specified uniquely by its (x, y) chromaticity coordinates and its luminance Y , and is often used to describe a color since the tristimulus values are straightforward to derive from the (x, y, Y) values. The biggest advantage of the (x, y) chromaticity coordinates is that they specify a magnitude-independent hue and purity of a color. A chromaticity diagram (see Fig. 2) is typically used to specify a color using its chromaticity coordinates. In Fig. 2, the horseshoe-shaped locus denotes the locus of monochromatic stimuli visible to the standard 2° observer (the gamut of visible colors). Shorter wavelength stimuli (starting at 380 nm, eliciting a relatively strong blue response) reside in the lower left of this horseshoe shape while the longer wavelengths (ending at 830 nm, eliciting a relatively strong red response) reside on the lower right, with the top of the horseshoe curve around 520 nm (eliciting a strong green response). The line connecting the blue and red corners is referred to as the *line of purples*. Colors on this line, although on the border of the gamut, have no counterpart in monochromatic sources of light and hence have no wavelengths associated with them.

The (x, y) chromaticity diagram is perceptually nonuniform: Unit vectors in the chromaticity space do not correspond to a unit change in perception even if the luminance is kept constant. In an attempt to improve the uniformity of the chromaticity diagram, in 1976, the CIE published a uniform chromaticity scale (UCS) diagram that scaled and normalized the XYZ

Color Spaces, Fig. 1 (a) RGB color matching functions (b) XYZ color matching functions including Judd-Vos modifications

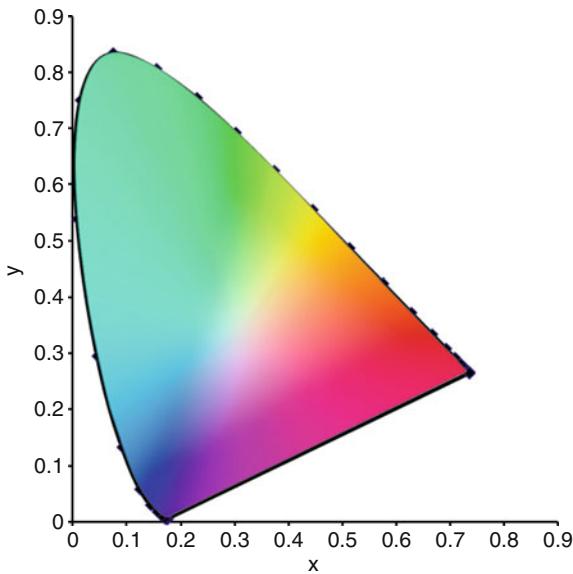


tristimulus values [6]. This chromaticity diagram is denoted by u' , v' axes, which are related to the XYZ tristimulus values by the following equations:

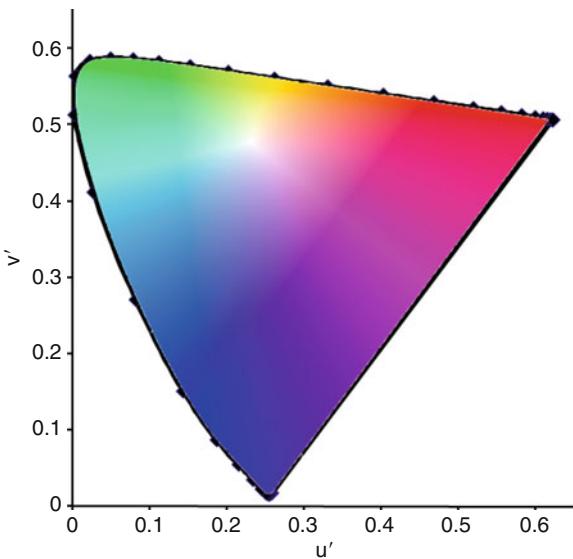
$$\begin{aligned} u' &= \frac{4X}{X + 15Y + 3Z} \\ v' &= \frac{9Y}{X + 15Y + 3Z}. \end{aligned} \quad (3)$$

Figure 3 shows the UCS. Just as in the case of Fig. 2, in this figure, the horseshoe-shaped locus represents the gamut of visible colors.

The CIEXYZ color space does not have perceptual correlates that would make it useful for common use. In an attempt to add perceptual behavior to color spaces, based on earlier works of many researchers, the CIE proposed a lightness scale along with two



Color Spaces, Fig. 2 CIE xy chromaticity diagram for a 2° observer



Color Spaces, Fig. 3 CIE UCS $u'v'$ chromaticity diagram for a 2° observer

chromatic scales. The CIELAB color space is one such space, where the axes are denoted by L^* (Lightness), a^* (redness–greenness), and b^* (yellowness–blueness). For a stimulus given by a tristimulus value X , Y , and Z , the CIELAB coordinates are given by:

$$L^* = 116 f(Y/Y_n) - 16$$

$$\begin{aligned} a^* &= 500 [f(X/X_n) - f(Y/Y_n)] \\ b^* &= 200 [f(Y/Y_n) - f(Z/Z_n)], \quad \text{where} \\ f(t) &= t^{1/3}, \quad \text{for } t > 0.008856 \\ f(t) &= 7.787 t + 16/116 \quad \text{otherwise.} \end{aligned} \quad (4)$$

In the above equations, the subscript n denotes the tristimulus values corresponding to the reference white chosen, which makes the CIELAB color space a relative color space. Given the CIELAB coordinates in a three-dimensional space, correlates of chroma and hue may be derived as follows:

$$C_{ab}^* = (a^{*2} + b^{*2})^{1/2} \quad (5)$$

$$h_{ab}^* = \tan^{-1}(b^*/a^*). \quad (6)$$

Under highly controlled viewing conditions, a CIELAB ΔE difference of 1 correlates with a single just noticeable difference in color. It is to be noted though that the CIELAB color difference measure was designed for color differences between uniform color patches in isolation.

In a similar construct, the CIE also recommended a CIELUV color space, based on the uniform chromaticity scale (UCS). This uses a subtractive shift from the reference white instead of the normalization based on division that is used in the CIELAB space. The equations to transform a tristimulus value from u^*, v^* coordinates to CIELUV are given by:

$$\begin{aligned} L^* &= 116 f(Y/Y_n) - 16 \\ u^* &= 13 L^*(u' - u'_n) \\ v^* &= 13 L^*(v' - v'_n), \quad \text{where} \\ f(t) &= t^{1/3}, \quad \text{for } t > 0.008856 \\ f(t) &= 7.787 t + 16/116 \quad \text{otherwise.} \end{aligned} \quad (7)$$

The u', v' coordinates for a tristimulus value are computed using (Eq. 3). As in the CIELAB definitions, the subscript n denotes the u', v' coordinates of the reference white being used. The descriptions of the u^* and v^* axes are similar to those in CIELAB: approximating redness–greenness and yellowness–blueness directions.

Based on these correlates, the CIE recommends that color difference measures in the two uniform-perception spaces CIELAB and CIELUV be given by the Euclidean difference between the coordinates of two color samples:

$$\begin{aligned}\Delta E^*_{ab} &= \left[(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2 \right]^{1/2} \\ \Delta E^*_{uv} &= \left[(\Delta L^*)^2 + (\Delta u^*)^2 + (\Delta v^*)^2 \right]^{1/2}\end{aligned}\quad (8)$$

where the differences are given between the corresponding color coordinates in the CIELAB and CIELUV spaces between the standard and test samples.

Many improvements to this basic color difference measure have been proposed and adopted over the years involving scaling the lightness, chroma, and hue differences appropriately based on the application and the dataset of samples to which the color difference measure has been adapted or improved [7]. Typically, color difference thresholds are dependent on application and thresholds for *perceptibility* judgments are significantly lower than thresholds for *acceptability* judgments. These color spaces were designed for threshold color differences and their application to supra-threshold (larger than about 5 units of ΔE) color differences is to be handled with care [17]. Many other color difference measures have been proposed and more recently, the CIE DE2000 has been adopted as a measure of color difference, again for uniform color patches under highly controlled viewing conditions, and is slowly gaining acceptance [7, 15].

These color spaces provide a powerful tool to model and quantify color stimuli and are used both in color difference modeling for color patches and, as well, have more recently been used to describe color appearance (see [19]). Models based on describing colors based on lightness, chroma, and hue are powerful in their abilities to enable communication of color stimuli, as well.

Color spaces for colorimetric analysis have the advantage that the need for communicating these colors to others can be done without much ambiguity, if the viewing conditions (reference white) are specified and well controlled. Arguably, the biggest disadvantage is that it is not straightforward to understand what a certain color coordinate in a certain color space would mean to an observer without having to use mathematical equations.

Color Spaces for Editing

Color spaces for editing are specifically designed with the following key requirements:

- Colors have perceptual correlates that are easily understood, such as hue, saturation, brightness, purity, etc.
- Colors described in these spaces are reproducible – within reason – across different media: monitors, printers, etc.
- Colors in these spaces are defined based on primaries – the axes:
 - On the display side of the world of color applications, these spaces are additive, defined by red, green, and blue primaries: Equal amounts of red and green will give yellow.
 - On the printing side, primaries are defined roughly by cyan, magenta, yellow, and black, and these spaces are subtractive: Knowing the print surface, equal densities of cyan and yellow will give green.

A Brief Note on “Gamma”

If a color system is linear, then for additive combinations of colors, a unit input of a color corresponds to a globally constant unit of the output signal; whereas for nonlinear combinations, a transfer function would determine the input–output relationship. This nonlinearity is often used in digital systems with limited available bits and in signal compression systems to take advantage of the available output signal bit depth by stretching small input codes over a larger range of output codes and compressing the larger input codes into a smaller output dynamic range. This is referred to as *gamma encoding*.

From an encoding perspective, in its simplest form, the input–output relationship is typically given by a gain-offset-gamma model, given by:

$$y = \text{round} \left[\left(2^N - 1 \right) \left(\alpha x + \beta \right)^\gamma \right] \quad (9)$$

where α denotes a scalar gain, N denotes the number of bits in a system, β is a scalar offset, γ denotes a power law with values larger than 1 (typically around 2.2), and x and y denote the normalized input and output signals respectively [1]. In encoding systems, the three channels typically have the same parameters α , β , γ , N . Display systems based on cathode ray tubes (CRTs) have an inherent response that follows the inverse relationship – large steps in input signal at the low end of the input signal cause a small change

in output whereas at the upper end of the signal range, small steps caused large output changes. It so happens that gamma encoding (using a power law of γ on the linear luminance input) the input prior to transmitting the data to a CRT display causes the display luminance to follow similar steps resulting in a net unity transfer function. This is also a useful means of encoding data to maximize bit-depth usage while reducing visibly apparent contouring on the output data and display [10, 18]. In the case of a quantized color space, for reasons of perceptual uniformity, it is preferable to establish a nonlinear relationship between color values and intensity or luminance.

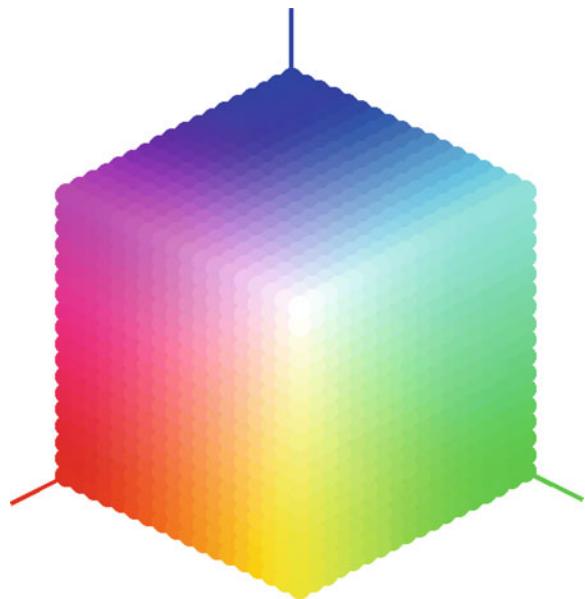
RGB Color Model

The specification of a color in the RGB color space implies a linear (after gamma is removed) or nonlinear (gamma applied) combination of the red, green, and blue primaries in varying strengths. In RGB color spaces, the manner in which colors are reproduced varies from device to device. For example, a color specified as an RGB triplet is more than likely going to look different from one display device to another when the exact same RGB triplet is provided as input, due to differences in the “color” of the primaries and also differences in gamma curves. This makes the RGB space a device-dependent color space. Specifying colors in device-dependent color spaces, although not preferred from a color-reproduction perspective, is often resorted to due to its ease in comprehension. An RGB color model can represent any color within an RGB color cube, as shown in Fig. 4. This color model is most commonly used in display applications where data is additive in nature.

For example, a full-strength yellow color is specified by (1.0, 1.0, 0.0), denoting the use of the red and green primaries at full strength and the blue primary completely turned off. In an 8-bit system, this will correspond to a code value of (255,255,0). A three-primary display with three independent color primaries (typically denoted by their CIE x,y chromaticity values along with that of white) is specified by:

$$\begin{bmatrix} x_R & x_G & x_B & x_W \\ y_R & y_G & y_B & y_W \end{bmatrix}. \quad (10)$$

An introduction to how specifications such as those in (Eq. 10) may be used to generate a transformation



Color Spaces, Fig. 4 RGB color cube commonly used in display applications

matrix from RGB to XYZ for linear RGB is given in the book [18].

Different RGB color spaces differ in their gamma values and the chromaticity coordinates in (Eq. 10). A useful compilation of various RGB color spaces may be found in a Web site hosted by Bruce Lindbloom [11].

A color mixing matrix for additive colors mixing is shown in Table 1, stating, for example, that a cyan color would be created using maximum intensity of green and blue primaries and none of the red primary.

CMY/CMYK Color Model

Printers, on the other hand, create colors using inks that are deposited on paper, in which case the manner in which they create color is called subtractive color mixing. The inks selectively absorb wavelengths of incident light and reflect the remainder. As a beam of light passes through an absorbing medium, the amount of light absorbed is proportional to the intensity of the incident light times the coefficient of absorption (at a given wavelength). This is often referred to as *Beer-Lambert-Bouguer law* and is given by:

$$A(\lambda) = \log_{10} \varepsilon(\lambda)c(\lambda)l(\lambda) \quad (11)$$

**Color Spaces, Table 1** Color mixing matrix for additive primaries

Primary used	Color displayed							
	Red	Green	Blue	Cyan	Yellow	Magenta	White	Black
Red	1	0	0	0	1	1	1	0
Green	0	1	0	1	1	0	1	0
Blue	0	0	1	1	0	1	1	0

where $\varepsilon(\lambda)$ denotes absorptivity, $c(\lambda)$ denotes the concentration, and $l(\lambda)$ denotes the path length for the beam of light. Stated differently, the higher the concentration or thickness or absorptivity of a certain absorptive material, the higher is absorption – the intensity of reflected or transmitted beam of light will be reduced [20]. The simplest model for printer inks is called the *block-dye model*, an idealized system where, for example, cyan ink absorbs all the red but none of the green or blue, with rectangular shapes for absorptivity as a function of wavelength.

In a subtractive-color setup, different thicknesses of the three primary inks may be deposited on top of each other to result in a final color to the observer. The colorant amounts required to print a stimulus designated by RGB emissions are given by $Y = 1 - X$, where $Y \in \{C, M, Y\}$ and $X \in \{R, G, B\}$, all normalized to unity. Real primary inks, however, do not correspond to these ideal functions and, hence, more sophisticated models need to include not just the spectral absorptions/reflectances of the inks, but the density (or area) of the inks and the characteristics of the media (paper) involved. The Kubelka-Munk equations describe the absorption and scattering of light as it passes through layers of ink and the substrate, for example, paper. Various extensions are used in practice that account for the shortcomings of the basic Kubelka-Munk analysis, considering issues such as nonlinearities in ink deposition, interactions between inks, etc., [8, 20]. In subtractive color mixing, the primaries are typically cyan (C), yellow (Y), and magenta (M). The color mixing matrix for subtractive color is shown in Table 2.

Much like RGB color systems, where the reference white made a difference in the appearance of a certain color, depending upon the kind of paper and inks used for printing, the rendered color can be significantly different from one printer to another.

Most printers use a “K” channel, denoting black ink, primarily because a black generated by mixing cyan, yellow, and magenta is not black enough in appearance. Additionally, to complicate matters, in order to print black, a printer would need to lay cyan,

magenta, and yellow inks on top of each other, making ink drying a cause for concern and additionally the limits of ink absorption by the substrate, for example, paper. Additionally, using 1 unit of black ink instead of 1 unit each of cyan, yellow, and magenta inks can lead to significant cost savings.

HSL/HSV Color Model

In order to make the representation of colors intuitive, colors may be ordered along three independent dimensions corresponding to the perceptual correlates of lightness, hue, and chroma. In device-dependent color spaces, there are many commonly used variants of these perceptual correlates: HSV is by far the most common. H stands for the perceptual correlate of hue; S stands for the saturation of a color, defined by the chroma of a color divided by its luminance (the more desaturated the color the closer it is to gray); and V stands for value (a perceptual correlate of lightness). This color model is commonly used in image processing and editing software. However, the HSV color model has two visualization representations, one of which is a cylinder with black at the bottom and pure full-intensity colors on the top, and the other is a representation by a cone, with black at the apex and white on the base. The equations used to convert RGB data into the HSV color space are given by:

$$V = \max \quad (12)$$

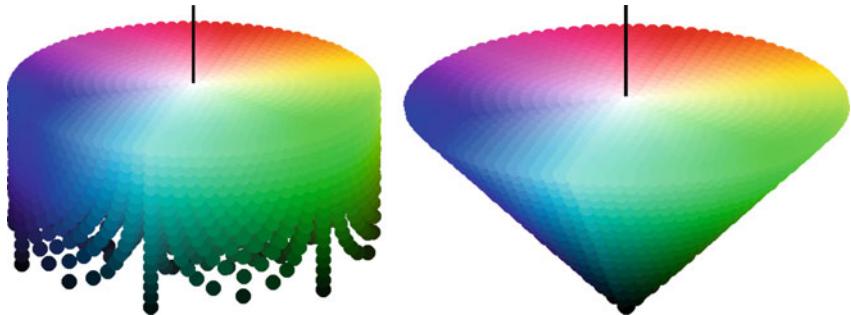
$$S = \begin{cases} 0 & \text{if } V = 0 \\ ((V - \min)/V) & \text{if } V > 0 \end{cases} \quad (13)$$

$$H = \begin{cases} 0 & \text{if } S = 0 \\ 60(G - B)/(max - min) & \text{if } (max = R \text{ and } G \geq B) \\ 60(G - B)/(max - min) + 360 & \text{if } (max = R \text{ and } G < B) \\ 60(B - R)/(max - min) + 120 & \text{if } max = G \\ 60(R - G)/(max - min) + 240 & \text{if } max = B \end{cases} \quad (14)$$

where max and min denote the maximum and minimum of the (R, G, B) triplet. These two representations are shown in Fig. 5. From the figure, it is apparent that saturation is not dependent on the

Color Spaces, Table 2 Color mixing matrix for subtractive primaries

Primary used	Color displayed							
	Red	Green	Blue	Cyan	Yellow	Magenta	White	Black
Cyan	0	1	1	1	0	0	0	1
Yellow	1	1	0	0	1	0	0	1
Magenta	1	0	1	0	0	1	0	1

Color Spaces, Fig. 5 The HSV color model represented as a cylinder and as a cone

intensity of the signal. It is, however, often useful in image processing applications to have an indicator of saturation given by a function of the intensity of the signal, resulting in a conical-shaped HSV color space (Fig. 5). When the conical representation is preferred, S is given by $(\max - \min)/(2^N - 1)$ where $2^N - 1$ denotes the largest possible value for R, G, or B. Other variants of the HSV color space also exist and are used as an intuitive link to RGB color spaces (HSB, or HLS denoting various correlates of hue, saturation, and brightness/lightness).

Other Color Spaces

Color spaces designed for editing and communication needs are typically formulated such that colors are encoded/transmitted in the color space of a reference device. Colors spaces fitting such a description include the sRGB color space, the YCC, YUV, YIQ color transmission spaces, the SWOP CMYK color space, Adobe RGB, and ProPhoto RGB, to list a few.

A popular mechanism to standardize colors across electronic devices such as printers, monitors, and the Internet is the use of the sRGB color space. Originally, this was proposed by Hewlett-Packard and Microsoft, and was later standardized by the International Electrotechnical Commission under IEC 61966-2-1 [12]. There are two primary parts to the sRGB standard: the viewing conditions and the necessary colorimetric definitions and transformations. The sRGB reference viewing environment corresponds to

conditions typical of monitor display viewing conditions and thus may not be as well suited for print material, due to the various proprietary gamut mapping algorithms in most printers that take advantage of each printer's color gamut. The colorimetric definitions provide the transforms necessary to convert between the sRGB color space and the CIEXYZ tristimulus color space as defined for a standard 2° observer. More specifically, the standard is written for a standard reference monitor that has Rec. 709 primaries and a D65 white point. An overview of the technical advantages and challenges of the sRGB color space may be found in Refs. [21, 26]. As was mentioned earlier, color spaces for video directly make use of the gamma-corrected signals, denoted R' , G' , B' , from camcorders, without any attempt to correlate to the linear signals used in color science, such as those in (Eq. 1). For still imaging as well as video, this problem can be mitigated by the use of the transform built into the sRGB standard, which includes a function for transforming from nonlinear signals I' to linear ones. On a scale of 0.0–1.0, for each of $I = R, G, B$, the following function is applied:

$$\begin{cases} I = I'/12.92, & \text{if } I' < 0.04045; \\ I = ((I' + 0.055)/1.055)^{2.4} & \text{otherwise.} \end{cases} \quad (15)$$

In the video industry, a common mode of communication is the YCbCr color space (YPbPr in the



analog domain) that converts RGB signal information into an opponent luma-chroma color space. A nonlinear transfer function is applied to linear-light R, G, B values and a weighted sum of the resulting R', G', B' values is used in the Y, Cb, and Cr signals. In the television domain, these signals have dynamic ranges (on an 8-bit scale) of 16–235 for the luma signal and 16–240 in the Cb and Cr signals. This is to allow for signal noise and potential signal processing noise, giving some head- and foot-room. The weights are different depending upon the color space that the data is being created for. For example, encoding R', G', B' signals with a 16–235 dynamic range into a color space defined by the NTSC primaries (often referred to as ITU-R BT.601), is given by:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \quad (16)$$

whereas when using HDTV (referred to as ITU-R BT.709) primaries, is given by:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.213 & 0.715 & 0.072 \\ -0.117 & -0.394 & 0.511 \\ 0.511 & -0.464 & -0.047 \end{bmatrix} \begin{bmatrix} R' \\ G' \\ B' \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}. \quad (17)$$

A useful reference for further details on this topic is a book by Keith Jack [14]. The Y channel typically contains most of the information in the image, as defined by spatial frequencies, and is hence sampled at much higher rates than the chroma signals. This greatly helps in the ability of the transmission system to compress luma and chroma data with low overheads when compared to luma-only systems. To aid compression formats, color images in the JPEG and JPEG2000 file formats also convert the R', G', B' information into the YCbCr color space prior to compression.

In the printing industry, a commonly specified color space is the SWOP (Specifications for Web Offset Publications) CMYK color space. The SWOP CMYK [2] is a proofing specification that has a well-established relationship between the CMYK input to a standard printer and its CIELAB values (an approximation of the perceptual coordinates of a color) and for a standardized dataset. Specifying images in the SWOP CMYK color space allows the printing house and the content creator to preview images on a common

baseline prior to printing. Most image editing software available nowadays allows the user to preview images in the SWOP CMYK color space.

Open Problems

Depending upon the application, color spaces have their individual, optimized uses. Device-independent color models like the CIEXYZ, CIELAB, CIELUV, and their derivatives are used most often to communicate color either between devices or between different color processing teams across the world. The International Color Consortium (ICC) has been extremely successful in standardizing device-independent color spaces between displays, printers, and capture devices [9, 13]. The color profiles that are stored and communicated in ICC profiles use an intermediate profile connection space (PCS) such as CIEXYZ or CIELAB. ICC profiles also store color transformation profiles to and from different color devices (say, from an input device such as a scanner to CIEXYZ, and from CIEXYZ to an output device such as a printer). For example, an sRGB ICC profile incorporates the color space transform from sRGB to the PCS and a SWOP CMYK ICC profile would incorporate the color space transform from the PCS to the CMYK output color space for a printer. Furthermore, depending upon the rendering intent (how the colors need to be represented on the output device), different transformations may be specified in the ICC profile. Interested readers are referred to more detailed discussions of this subject such as the comprehensive books by Hunt [10], BillMeyer and Saltzman [1], Kuehni [16], Fairchild [19], Sharma [23], and Green and MacDonald [8].

References

1. Berns RS (2000) Billmeyer and Saltzman's principles of color technology, 3rd edn. Wiley, New York
2. CGATS TR 001 (1995) Graphic technology – color characterization data for type 1 printing. American National Standards Institute, Washington, DC
3. Commission Internationale de l'Eclairage (1926) The basis of physical photometry, CIE proceedings 1924. Cambridge University Press, Cambridge
4. Commission Internationale de l'Eclairage (1931) Proceedings international congress on illumination. Cambridge University Press, Cambridge

5. Commission Internationale de l'Eclairage (1964) CIE proceedings (1964) Vienna session, committee report E-1.4.1. Bureau Central de la CIE, Paris
6. Commission Internationale de l'Eclairage (1986) CIE publication 15.2, colorimetry. Central Bureau CIE, Vienna
7. Commission Internationale de l'Eclairage (2004) Colorimetry, publication CIE 15:2004, 3rd edn. CIE, Vienna
8. Green P, MacDonald L (eds) (2002) Colour engineering: achieving device independent colour. Wiley, Chichester
9. <http://www.color.org>. International Color Consortium
10. Hunt RWG (2004) The reproduction of color, 6th edn. Wiley, Chichester/Hoboken
11. Information About RGB Working Spaces. <http://www.brucelindbloom.com/WorkingSpaceInfo.html>
12. International Electrotechnical Commission (1999) IEC 61966-2-1: multimedia systems and equipment – colour measurement and management – Part 2-1: colour management – default RGB colour space – sRGB. IEC, Geneva
13. International Organization for Standardization (2005) ISO 15076-1, image technology colour management – Architecture, profile format and data structure – Part 1:Based on ICC.1:2004-10, ISO, Geneva
14. Jack K (2001) Video demystified – a handbook for the digital engineer, 3rd edn. LLH Technology Publishing, Eagle Rock
15. Kuehni RG (2002) CIEDE2000, milestone or a final answer? Color Res Appl 27(2):126–127
16. Kuehni RG (2003) Color space and its divisions: color order from antiquity to the present. Wiley, Hoboken
17. Kuehni RG (2004) Color: an introduction to practice and principles, 2nd edn. Wiley, Hoboken
18. Li Z-N, Drew MS (2004) Fundamentals of multimedia. Prentice-Hall, Upper Saddle River
19. Fairchild MD (2005) Color appearance models, 3rd edn. <http://www.wiley.com/WileyCDA/WileyTitle/productCd-1119967031.html>
20. McDonald R (1997) Colour physics for industry, 2nd edn. Society of Dyers and Colourists, Bradford
21. Microsoft Corporation (2001) Colorspace interchange using srgb. <http://www.microsoft.com/whdc/device/display/color/sRGB.mspx>
22. Nassau K (1983) The physics and chemistry of color: the fifteen causes of color. Wiley, New York
23. Sharma G (ed) (2003) Digital color imaging handbook. CRC, Boca Raton
24. Speranskaya NI (1959) Determination of spectrum color coordinates for twenty-seven normal observers. Opt Spectrosc 7:424–428
25. Stiles WS, Burch JM (1959) Npl colour-matching investigation: final report (1958). Opt Acta 6:1–26
26. Stokes M, Anderson M, Chandrasekar S, Motta R (1996) A standard default color space for the internet: sRGB. <http://www.color.org/sRGB.html>
27. Wyszecki G, Stiles WS (2000) Color science: concepts and methods, quantitative data and formulae, 2nd edn. Wiley, New York

Color Specification Systems

► Color Model

Compressed Sensing

► Compressive Sensing

Compressive Sensing

Aswin C. Sankaranarayanan¹ and Richard G. Baraniuk²

¹ECE Department, Rice University, Houston, TX, USA

²Department of Electrical and Computer Engineering, Rice University 2028 Duncan Hal, Houston, TX, USA

Synonyms

Compressed sensing

Related Concepts

► Dimensionality Reduction

Definition

Compressed sensing refers to parsimonious sensing, recovery, and processing of signals under a sparse prior.

Background

The design of conventional sensors is based heavily on the Shannon-Nyquist sampling theorem which states that a signal x band limited to W Hz is determined completely by its discrete time samples provided the sampling rate is greater than $2W$ samples per second. This theorem is at the heart of modern signal processing as it enables signal processing in the discrete time or digital domain without any loss of information. However, for many applications, the Nyquist sampling rate is high as well as redundant and unnecessary. As a motivating example, in modern cameras, the high resolution of the CCD sensor reflects the large amount of data sensed to capture an image. A 10 megapixel camera, in effect, takes 10 million linear measurements

of the scene. Yet, almost immediately after capture, redundancies in the image are exploited to compress the image significantly, often by compression ratios of 100:1 for visualization and even higher for detection and classification tasks. This suggests immense wastage in the overall design of the conventional camera.

Compressive sensing (CS) refers to a sampling paradigm where additional structure on the signal is exploited to enable sub-Nyquist sampling rates. The structure most commonly associated with CS is that of signal sparsity in a transform basis. As an example, the basis behind most image compression algorithms is that images are sparse (or close to sparse) in transform bases such as wavelets and DCT. In such a scenario, a CS camera takes *under-sampled* linear measurements of the scene. Given these measurements, the image of the scene is recovered by searching for the image that is sparsest in the transform basis (wavelets or DCT) while simultaneously satisfying the measurements. This search procedure can be shown to be convex. Much of CS literature revolves around the design of linear measurement matrices, characterizing the number of measurements required and the design of image/signal recovery algorithms.

Theory

Compressive sensing [1–3] enables reconstruction of sparse signals from under-sampled linear measurements. A vector \mathbf{s} is termed K sparse if it has at most K nonzero components, or equivalently, if $\|\mathbf{s}\|_0 \leq K$, where $\|\cdot\|_0$ is the ℓ_0 norm or the number of nonzero components. Consider a signal (e.g., an image or a video) $\mathbf{x} \in \mathbb{R}^N$, which is sparse in a basis Ψ , that is, $\mathbf{s} \in \mathbb{R}^N$, defined as $\mathbf{s} = \Psi^T \mathbf{x}$, is sparse. Examples of the sparsifying basis Ψ for images include DCT and wavelets. The main problem of interest is that of sensing the signal \mathbf{x} from linear measurements. With no additional knowledge about \mathbf{x} , N linear measurements of \mathbf{x} are required to form an invertible linear system. In a conventional digital camera, an identity sensing matrix is used so that each pixel is sensed directly. For sensing reflectance fields, optimal linear sensing matrices have been designed [4].

The theory of compressive sensing shows that it is possible to reconstruct \mathbf{x} from M linear measurements even when $M \ll N$ by exploiting the sparsity

of $\mathbf{s} = \Psi^T \mathbf{x}$. Consider a measurement vector $\mathbf{y} \in \mathbb{R}^M$ obtained using an $M \times N$ measurement matrix Φ , such that

$$\mathbf{y} = \Phi \mathbf{x} + e = \Phi \Psi \mathbf{s} + e = \Theta \mathbf{s} + e, \quad (1)$$

where e is the measurement noise (see Fig. 1) and $\Theta = \Phi \Psi$. The components of the measurement vector \mathbf{y} are called the *compressive measurements* or compressive samples. For $M < N$, estimating \mathbf{x} from the linear measurements is an ill-conditioned problem. However, when \mathbf{x} is K sparse in the basis Ψ , then CS enables recovery of \mathbf{s} (or alternatively, \mathbf{x}) from $M = O(K \log(N/K))$ measurements, for certain classes of matrices Θ . The guarantees on the recovery of signals extend to the case when \mathbf{s} is not exactly sparse but compressible. A signal is termed compressible if its sorted transform coefficients decay according to power law, that is, the sorted coefficient of \mathbf{s} decay rapidly in magnitude [5].

Restricted Isometry Property (RIP)

The condition for stable recovery for both sparse and compressible signals is that the matrix Θ satisfies the following property. Given S -sparse vector \mathbf{s} , $\exists \delta_S$, $0 < \delta_S < 1$ such that

$$(1 - \delta_S) \|\mathbf{s}\|_2 \leq \|\Theta \mathbf{s}\|_2 \leq (1 + \delta) \|\mathbf{s}\|_2. \quad (2)$$

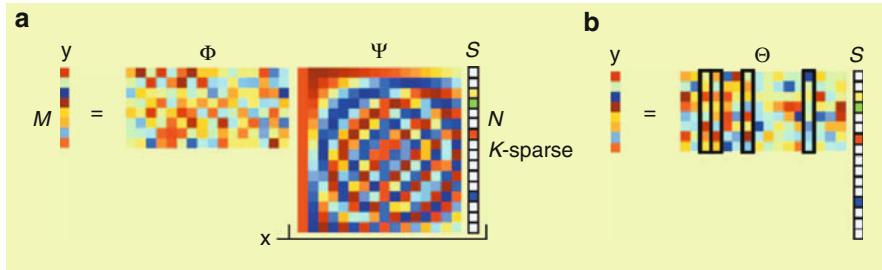
Stable recovery is guaranteed when (Eq. 2) is satisfied for $S = 2K$. This is referred to as the *Restricted Isometry Property* (RIP) [2, 7]. In particular, when Ψ is a fixed basis, it can be shown that using a randomly generated sub-Gaussian measurement matrix Φ , ensures that Θ satisfies RIP with a high probability provided $M = O(K \log(N/K))$. Typical choices for Φ (or equivalently, Θ) are matrices whose entries are independently generated using the Radamacher or the sub-Gaussian distribution.

Signal Recovery

Estimating K -sparse vectors that satisfy the measurement equation of (Eq. 1) can be formulated as the following ℓ_0 optimization problem:

$$(P0) : \min \|\mathbf{s}\|_0 \text{ s.t. } \|\mathbf{y} - \Phi \Psi \mathbf{s}\|_2 \leq \epsilon, \quad (3)$$

with ϵ being a bound for the measurement noise e in (Eq. 1). While this is a NP-hard problem in general, the equivalence between ℓ_0 and ℓ_1 norm for such



Compressive Sensing, Fig. 1 (a) Compressive sensing measurement process with a random Gaussian measurement matrix Φ and discrete cosine transform (DCT) matrix Ψ . The vector of coefficients s is sparse with $K = 4$. (b) Measurement process

systems [8] allows us to reformulate the problem as one of ℓ_1 norm minimization.

$$(P1) : \hat{s} = \arg \min \|s\|_1 \text{ s.t. } \|y - \Phi \Psi s\| \leq \epsilon. \quad (4)$$

It can be shown that, when $M = O(K \log(N/K))$, the solution to the $(P1) - \hat{s}$ – is, with overwhelming probability, the K -sparse solution to $(P0)$. In particular, the estimation error can be bounded as follows:

$$\|s - \hat{s}\|_2 \leq C_0 \|s - s_K\| / \sqrt{K} + c_1 \epsilon, \quad (5)$$

where s_K is the best K -sparse approximation of s .

There exist a wide range of algorithms that solve $(P1)$ to various approximations or reformulations [9, 10]. One class of algorithms model $(P1)$ as a convex problem and recast it as a linear program (LP) or a second order cone program (SOCP) for which there exist efficient numerical techniques. Another class of algorithms employ greedy methods [11] which can potentially incorporate other problem-specific properties such as structured supports [12].

Hardware Implementations

Compressive sensing has been successfully applied to sense various visual signals such as images [13, 14], videos [15] and light transport matrices [16]. The single-pixel camera (SPC) [13] for CS of images is designed as follows. The SPC consists of a lens that focuses the scene onto a digital micro-mirror device (DMD), which takes the place of the CCD array in

with $y = \Phi x$. There are four columns that correspond to nonzero s_i coefficients; the measurement vector y is a linear combination of these columns (Figure from [6])

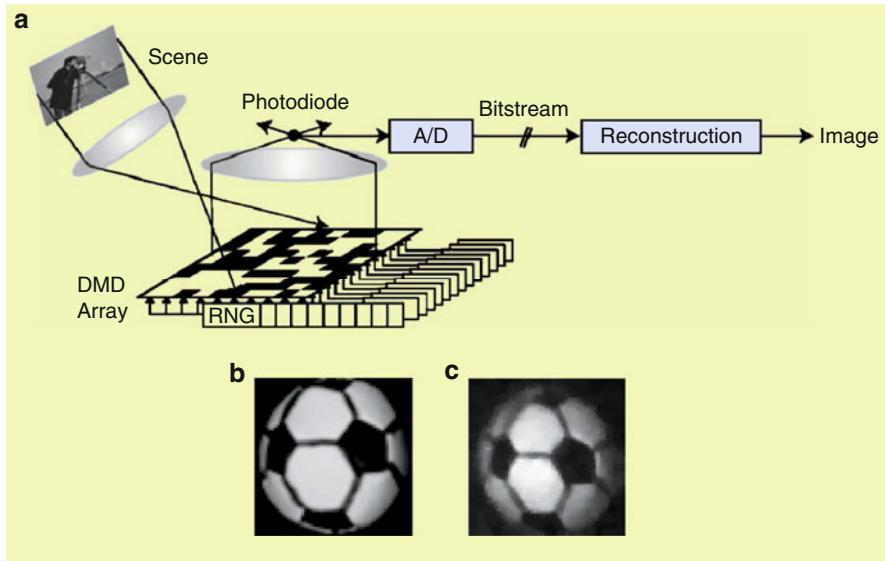
a conventional camera. Each micro-mirror can be oriented in one of two possible angles. Micro-mirrors that are in the same angle/direction are focused onto a photodetector (hence, the single pixel) using a second lens (see Fig. 2). Each measurement that is obtained corresponds to an inner product of the scene with a set of 1s and 0s corresponding to the state of the micro-mirrors. By flipping the micro-mirrors randomly according to a Bernoulli distribution, multiple compressive measurements are obtained.

The SPC is most useful when sensing is costly. One such example is in sensing at wavelengths beyond the visible spectrum, where exotic and costly materials are needed to detect light. For such applications, the cost of building a focal plane array of even VGA resolution is prohibitive. However, the SPC needs only a single photodiode tuned to the spectrum of interest, with otherwise minimal changes to the underlying architecture.

Implications for Computer Vision

Novel Sensors

Compressive sensing enables a new sensing design that measures random linear projection of a scene and subsequently reconstructs the image of the scene using these measurements. The reconstructed image is no different from one captured from a suitably placed camera, barring reconstruction artifacts. However, compressive cameras enable parsimony in sensing – which can be extremely useful in problems where sensing is costly. This could be due to storage,



Compressive Sensing, Fig. 2 (a) Single-pixel, compressive sensing camera. (b) Conventional digital camera image of a soccer ball. (c) 64×64 black-and-white image of the same ball

($N = 4,096$ pixels) recovered from $M = 1,600$ random measurements taken by the camera in (a). The images in (b) and (c) are not meant to be aligned (Figure from [6])

as in *the data deluge* problem in large sensor networks. Compressive sensing is extremely valuable in time sensitive applications such as medical imaging, sensing and modeling fluid dispersions, time varying reflectance fields, and in general, high speed imaging. Finally, sensing could be costly due to adverse affects caused by the process of sensing. In electron microscopy, there are fundamental limitations on the number of images of a thin layer of tissue that comes as the tissue is progressively destroyed in the process of imaging. In all such applications, where sensing is costly in some manner, CS offers better trade-offs over traditional linear sensing.

Tailoring Vision Algorithms for Compressive Data

In many cases, it is beneficial to work on the compressive measurements directly without reconstructing the images. As an example, in background subtraction, the silhouettes are spatially sparse and in many applications far sparser than the original images (in a transform basis). Given a static background image \mathbf{x}_s , silhouettes are recovered by identifying $(\mathbf{y} - \Phi\mathbf{x}_s)$ as compressive measurements of the silhouettes [17]. This can lead to silhouette recovery at extremely high compression ratios. An other example is in video CS for dynamic textures, where the linear dynamical

system parameters of the scene can be recovered by a suitably designed acquisition device [15].

Beyond Sparse Models

While the theory of CS relies on assumptions of sparsity or compressibility of the signal in a transform basis, it can be extended to signals that exhibit additional structures. One such idea is that of model-based CS [12] where in the signal exhibits sparsity in a space of models which encompasses models such as block sparsity, union of subspaces, and wavelet tree models. Another idea is in the use of non-sparse models such as Markov Random fields [18] for CS – with the eventual goal of using the Ising model for sensing images and background subtracted silhouettes. Such progress hints at interesting avenues for future research governed by the use of rich models in existing vision literature for the task of sensing.

Machine Learning and Signal Processing on Compressive Data

Restricted Isometry Property (see (Eq. 2)) for random matrices implies that, for K -sparse signals, distances are approximately preserved under random projections. A host of machine learning and signal processing algorithms depend on pairwise distances of points as

opposed to their exact location. For such algorithms, almost identical results can be obtained by applying them to randomly projected data as opposed to the data in the original space [19, 20]. This has tremendous advantages as the random projected data lies on a much lower dimensional space, and can be directly obtained from a compressive imager. Such ideas have been applied for detection and classification of signals in compressed domain [21].

References

1. Candès E, Romberg J, Tao T (2006) Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Trans Inf Theory* 52(2): 489–509
2. Candès E, Tao T (2006) Near optimal signal recovery from random projections: universal encoding strategies? *IEEE Trans Inf Theory* 52(12):5406–5425
3. Donoho D (2006) Compressed sensing. *IEEE Trans Inf Theory* 52(4):1289–1306
4. Schechner Y, Nayar S, Belhumeur P (2003) A theory of multiplexed illumination. International conference on computer vision, Nice, France, pp 808–815
5. Haupt J, Nowak R (2006) Signal reconstruction from noisy random projections. *IEEE Trans Inf Theory* 52(9):4036–4048
6. Baraniuk R (2007) Compressive sensing. *IEEE Signal Process Mag.* 24(4):118–120, 124
7. Davenport M, Laska J, Boufounos P, Baraniuk R (2009) A simple proof that random matrices are democratic. Technical report TREE 0906, Rice University, ECE Department
8. Donoho D (2006) For most large underdetermined systems of linear equations, the minimal ℓ_1 -norm solution is also the sparsest solution. *Comm Pure Appl Math* 59(6):797–829
9. Candès E, Tao T (2005) Decoding by linear programming. *IEEE Trans Inf Theory* 51(12):4203–4215
10. Tibshirani R (1996) Regression shrinkage and selection via the lasso. *J R Stat Soc B* 58(1):267–288
11. Needell D, Tropp J (2009) CoSaMP: iterative signal recovery from incomplete and inaccurate samples. *Appl Comput Harmon Anal* 26(3):301–321
12. Baraniuk R, Cevher V, Duarte M, Hegde C (2010) Model-based compressive sensing. *IEEE Trans Inf Theory* 56(4):1982–2001
13. Duarte M, Davenport M, Takhar D, Laska J, Sun T, Kelly K, Baraniuk R (2008) Single-pixel imaging via compressive sampling. *IEEE Signal Process Mag* 25(2):83–91
14. Wagadarikar A, John R, Willett R, Brady D (2008) Single disperser design for coded aperture snapshot spectral imaging. *Appl Opt* 47(10):B44–B51
15. Sankaranarayanan AC, Turaga P, Baraniuk R, Chellappa R (2010) Compressive acquisition of dynamic scenes. In: ECCV, Heraklion, Greece, pp 129–142
16. Peers P, Mahajan D, Lamond B, Ghosh A, Matusik W, Ramamoorthi R, Debevec P (2009) Compressive light transport sensing. *ACM Trans Graph* 28(1):1–3
17. Cevher V, Sankaranarayanan A, Duarte M, Reddy D, Baraniuk R, Chellappa R (2008) Compressive sensing for background subtraction. In: Proceedings of the European conference on computer Vision (ECCV), Marseille, France
18. Cevher V, Duarte M, Hegde C, Baraniuk R (2008) Sparse signal recovery using markov random fields. *Neural Info Proc Sys*, Vancouver, BC, Canada, pp 257–264
19. Davenport M, Boufounos P, Wakin M, Baraniuk R (2010) Signal processing with compressive measurements. *IEEE J Sel Top Signal Process* 4(2):445–460
20. Hegde C, Wakin M, Baraniuk R (2007) Random projections for manifold learning. Vancouver, BC
21. Davenport M, Duarte M, Wakin M, Laska J, Takhar D, Kelly K, Baraniuk R (2007) The smashed filter for compressive classification and target recognition, San Jose, CA

Computational Symmetry

Yanxi Liu

Computer Science and Engineering, Penn State University, University Park, PA, USA

Synonyms

[Symmetry-based X](#); [Symmetry detection](#)

Definition

Computational symmetry is a branch of research using computers to model, analyze, synthesize, and manipulate symmetries in digital forms, imagery, or otherwise [1].

Background

Symmetry is a pervasive phenomenon presenting itself in all forms and scales, from galaxies to microscopic biological structures, in nature and man-made environments. Much of one's understanding of the world is based on the perception and recognition of recurring patterns that are generalized by the mathematical concept of symmetries [2–4]. Humans and animals have an innate ability to perceive and take advantage of symmetry in everyday life [5–8], while harnessing this powerful insight for machine intelligence remains a challenging task for computer scientists.

Interested readers can find several influential symmetry-related papers below to gain a historic



perspective: the wonderful exposition on the role of symmetry in “Biological Shape and Visual Science” by Blum in 1973 [9]; in 1977, the “Description and Recognition of Curved Objects” reported by Nevatia and Binford, where bilateral symmetry of an object about different axes is examined [10]; the method of detecting angle/side regularities of closed curves and plateaus in one-dimensional patterns by Davis, Blumenthal, and Rosenfeld [11, 12]; the introduction of the term *skewed symmetry* by Takeo Kanade in 1981 [13]; the exposition on “Smoothed Local Symmetries and Their Implementation” by Brady and Asada [14]; the theory of *recognition-by-components* (RBC) proposed by Biederman in 1985 [15]; “Perceptual Grouping and the Natural Representation of Natural Form” using superquadrics as restricted *generalized cylinders* (GC) by Pentland in 1986 [16]; “Perceptual Organization and Visual Recognition” by Lowe [17], where the non-coincidental appearance of symmetry in the real world was noted; and the “Symmetry-Seeking Models for 3D Object Reconstruction” (1987) illustrated by Terzopoulos, Witkin, and Kass [18].

A computational model for symmetry is especially pertinent to computer vision, computer graphics, and machine intelligence in general because symmetry is:

- *Ubiquitous*: Both the physical and digital worlds are filled with various forms of symmetry, near-symmetry, and distorted symmetry patterns.
- *Essential*: Intelligent beings perceive and interact with the chaotic real world in the most efficient and effective manner by capturing its essential structures and substructures – the generators of symmetry, near-symmetry, distorted symmetry, and/or recurring patterns.
- *Compact*: The recognition of symmetries is the first step towards minimizing redundancy, often leading to drastic reductions in computation.
- *Aestho-physiological*: From a butterfly to an elephant, from a tea cup to a building, symmetry or deviation from it has been a time-honored principle for design (by nature or by human) that can guide machine perception, detection, recognition, and synthesis of the real world.

Figure 1 from [19] shows the statistics of published papers during the period of 1974–2009 (36 years) in several major computer vision/graphics conferences and journals. An increasing level of interests in computational symmetry, and a dominant role reflection

symmetry played in the past several decades, can be observed in both computer vision and computer graphics literature.

Theory

From the spirit of Felix Klein’s Erlangen program [20] that described geometry as the study of a space that is invariant under a given transformation group to the Gestalt principles of perception [21], symmetries and group theory play an important role in describing the geometry and appearance of an object. Informally, one may think of symmetry as expressing the notion that a figure or an object is made from multiple copies of smaller unit that are interchangeable. Mathematically, this notion is formalized by examining the effect of transformations on the object in a certain space such that its subparts map to each other.

More formally, in a metric space M , a *symmetry* $g \in G$ of a set $S \subseteq M$ is an isometry (a distance preserving transformation) that maps S to itself (an automorphism), $g(S) = S$. The transformation g keeps S invariant as a whole while permuting its parts. Symmetries G of S form a mathematical *group* $\{G, *\}$, closed under transformation composition $*$, called the *symmetry group* of S [3].

Basic Concepts

The basic definitions of symmetry and group theory can be found in most general mathematic textbooks on modern algebra. In particular the following books are recommended: *Geometry* by Coxeter [22], *Generators and relations for discrete groups* by Coxeter and Moser [3], *Symmetry* by Weyl [2], and *Tilings and Patterns* by Grünbaum and Shephard [23].

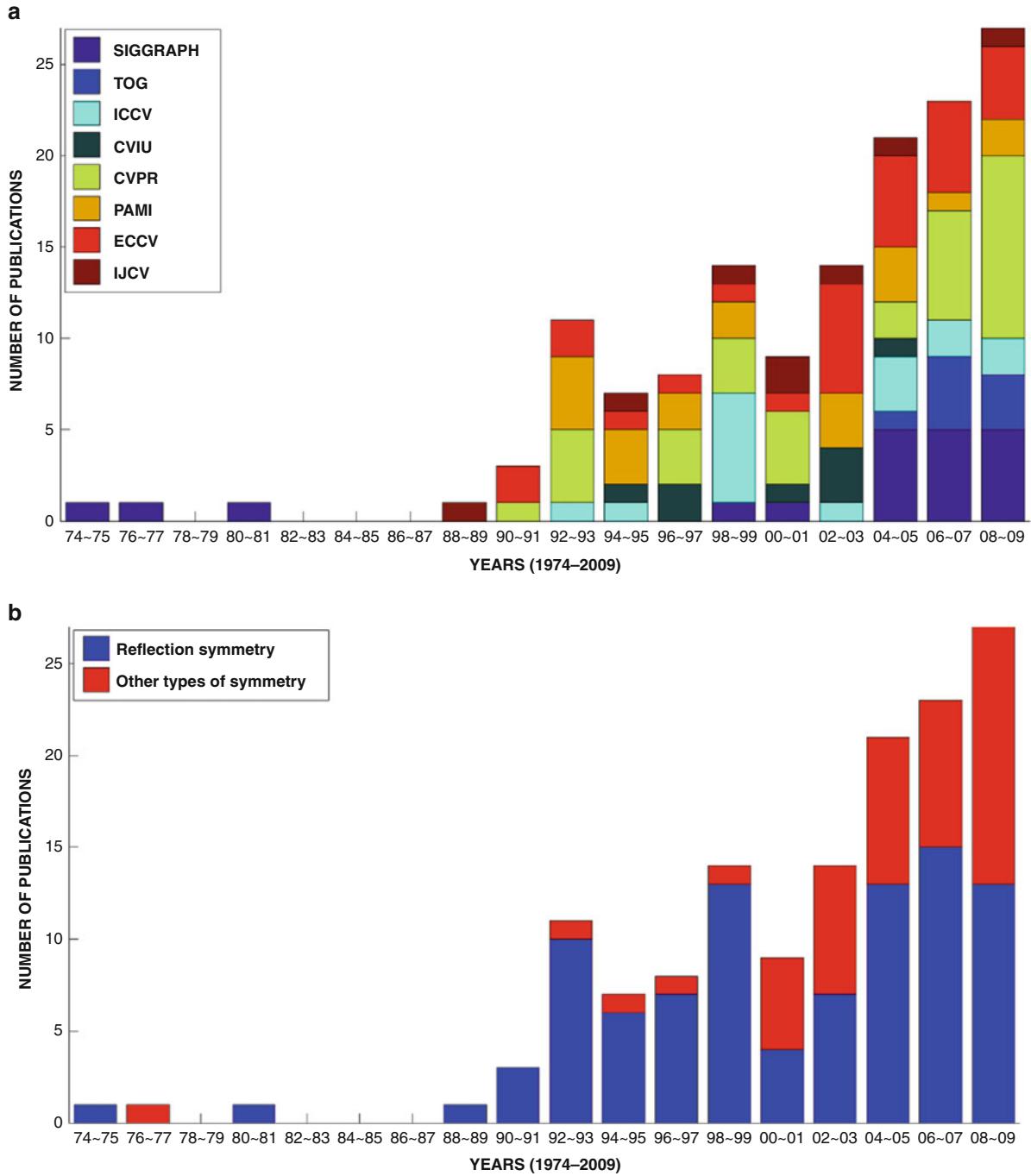
Some key concepts and computationally relevant definitions [19] are provided below:

Definition 1 Let S be a subset of R^n . Then an isometry (a distance preserving mapping) g is a symmetry of S if and only if $g(S) = S$.

Definition 2 A symmetry g for a set $S \in R^n$ is a primitive symmetry if and only if (1) $g \neq e$ where e is an identity mapping; and (2) if $g = g_1g_2$, $g_1 \neq e$, $g_2 \neq e$, then neither g_1 nor g_2 is a symmetry of S .

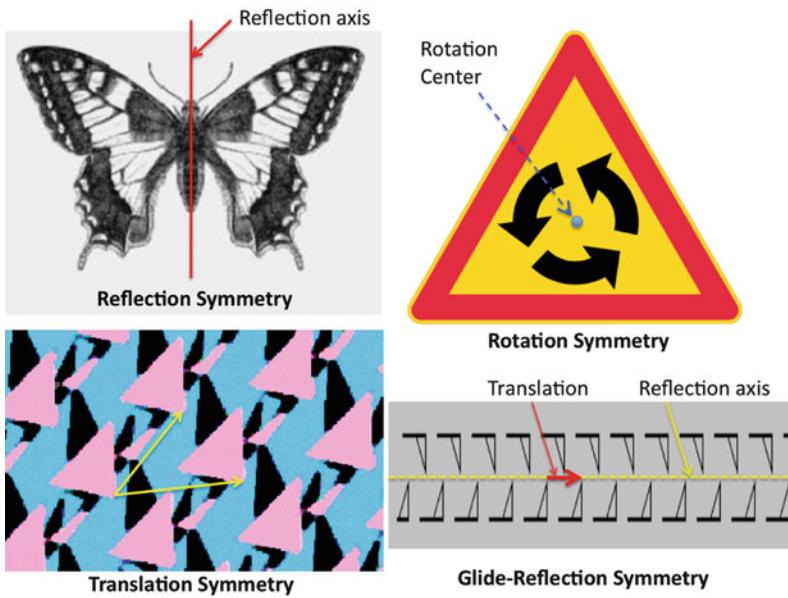
For example, in 2D Euclidean space R^2 , there are four types [2–4] of primitive symmetries $g(S) = S$.





Computational Symmetry, Fig. 1 From the publication statistics (data collected in [19]), it is obvious that (1) there is an increasing trend of interests in computational symmetry and (2) research on reflection symmetry has been dominating the field in the past several decades, though there is a recent growing awareness of the whole symmetry spectrum. A similar reflection-symmetry-dominating trend has also been

observed in the psychology literature for human perception of symmetries [8] (a) Publications on symmetry detection and applications in major computer vision and computer graphics conferences/journals (b) Dividing papers into reflection symmetry alone versus other types of symmetries (rotation, translation, and glide reflection)



Computational Symmetry, Fig. 2 An illustration of patterns with the four different primitive symmetries, respectively, in 2D Euclidean space. Reflection symmetry has the reflection axis as its point-wise invariance. Rotation symmetry has the *center* of rotation as its invariant point. An n -fold rotational symmetry with respect to a particular point (in 2D) or axis (in 3D)

means that, setwise, a rotation by an angle of $360/n$ does not change the object. Glide reflection is composed of a translation that is $1/2$ of the smallest translation symmetry t and a reflection r with respect to a reflection axis along the direction of the translation. There are no invariant points under translation and glide-reflection symmetries

They are, without loss of generality, for the four images $f(x, y)$ shown in Fig. 2:

1. *Reflection*: $f(x, y) = f(-x, y)$; its reflection axis (plane) remains invariant under the reflection.
 2. *Rotation*: $f(x, y) = f(r \cos(2\pi/n), r \sin(2\pi/n))$, $r = \sqrt{x^2 + y^2}$, n is an integer ($n = 3$ in the top right of Fig. 2), and its rotation center point (axis) remains invariant under the rotation.
 3. *Translation*: $f(x, y) = f(x + \Delta x, y + \Delta y)$, for some $\Delta x, \Delta y \in R$, no invariant points exist.
 4. *Glide Reflection*: $f(x, y) = f(x + \Delta x, -y)$, for some $\Delta x \in R$, no invariant points exist. A glide reflection g can be expressed as $g = tr$, where t is a translation and r is a reflection whose axis of reflection is along the direction of the translation.
- Note: neither t nor r alone is a symmetry of S ; thus, g is a primitive symmetry of S .

Definition 3 Let G be a nonempty set with a well-defined binary operation $*$ such that for each ordered pair $g_1, g_2 \in G$, $g_1 * g_2$ is also in G . $(G, *)$ is a **group** if and only if:

1. There exists an *identity element* $e \in G$ such that $e * g = g = g * e$ for all $g \in G$.

2. Any element g in G has an *inverse* $g^{-1} \in G$ such that $g * g^{-1} = g^{-1} * g = e$.
3. The binary operation $*$ is associative: $a * (b * c) = (a * b) * c$ for all $a, b, c \in G$.

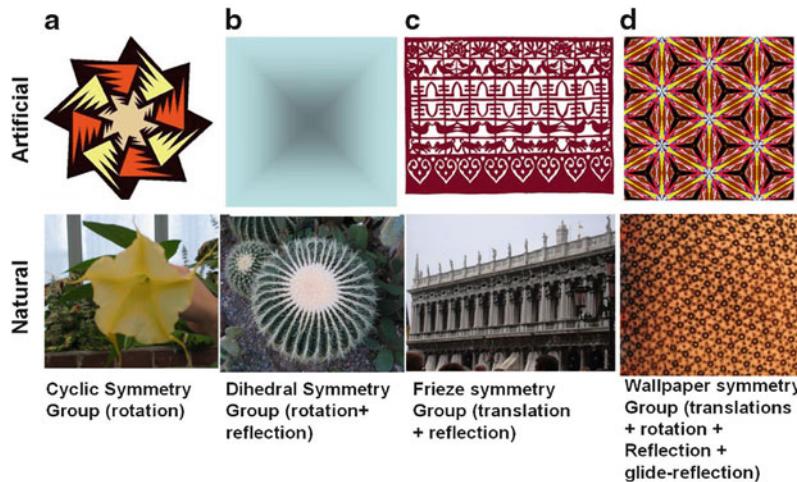
Using the composition of transformations on R^n as the binary operation $*$, one can prove that *symmetries of a subset $S \subset R^n$ form a group*, which is called the *symmetry group* of S . Figure 3 and Table 1 illustrate the four distinct types of symmetry groups in R^2 .

Definition 4 All the symmetries of R^n form the **Euclidean group** \mathcal{E} .

Proposition 5 Symmetries of a subset $S \subset R^n$ form a symmetry group G_S of S where $G_S \subset \mathcal{E}$ (see a proof in [24]).

Definition 6 A **point group** G is a symmetry group that leaves a single point fixed. In other words, $G = G_x$, here G_x is the stabilizer subgroup on x .

Definition 7 A **space group** is a group G of operations which leaves the infinitely extended, regularly repeating lattice pattern $L \subset R^n$ invariant while for all nontrivial $g \in G$, i.e., $g \neq e$, and $x \in R^n$, $g(x) \neq x$.



Computational Symmetry, Fig. 3 Sample images of symmetry categorized by their respective ideal symmetry groups: column (a) cyclic group containing rotation symmetries only, (b) dihedral group (reflection and rotation), (c) frieze group (translation plus reflection and rotation), and (d) wallpaper group

(translation, rotation, reflection, and glide-reflection). The *top row* contains synthetic patterns, while the *bottom row* shows photos of real-world scenes (*bottom right* is an image of a transverse slice of skeletal muscle magnified with a transmitter microscope 800,000 times)

Computational Symmetry, Table 1 Discrete symmetry groups in R^2 (See Fig. 3 for sample images)

Name	Group type	Symbol	Order	Primitive symmetry	Example
Cyclic	Point	C_n	n	Rotation	Fig. 3a
Dihedral	Point	D_n	$2n$	Rotation and reflection	Fig. 3b
Frieze	Space	G_{frieze}	∞	All four 2D-primitive symmetry types	Fig. 3c
Wallpaper	Space	$G_{\text{wallpaper}}$	∞	All four 2D-primitive symmetry types	Fig. 3d

Crystallographic Groups

An n -dimensional periodic pattern is formed by repeating a unit pattern in equal intervals along $k \leq n$ directions. A mature mathematical theory for symmetries of periodic patterns in n -dimensional Euclidean space has been known for over a century [25–28], namely, the *crystallographic groups* [2, 22, 23]. An important mathematical discovery is the answer to the first part of the Hilbert’s 18th problem [29]: regardless of dimension n and despite an infinite number of possible instantiations of periodic patterns, the number of distinct symmetry groups for periodic patterns in any Euclidean space R^n is always *finite*. For 2D monochrome patterns, there are 7 *frieze-symmetry groups* translating along one direction (strip patterns) [22] (Fig. 3c) and 17 *wallpaper groups* covering the whole plane [23, 25] (Fig. 3d). In 3D, there are 230 different *space groups* [30] generated by three linearly independent translations (regular crystal patterns).

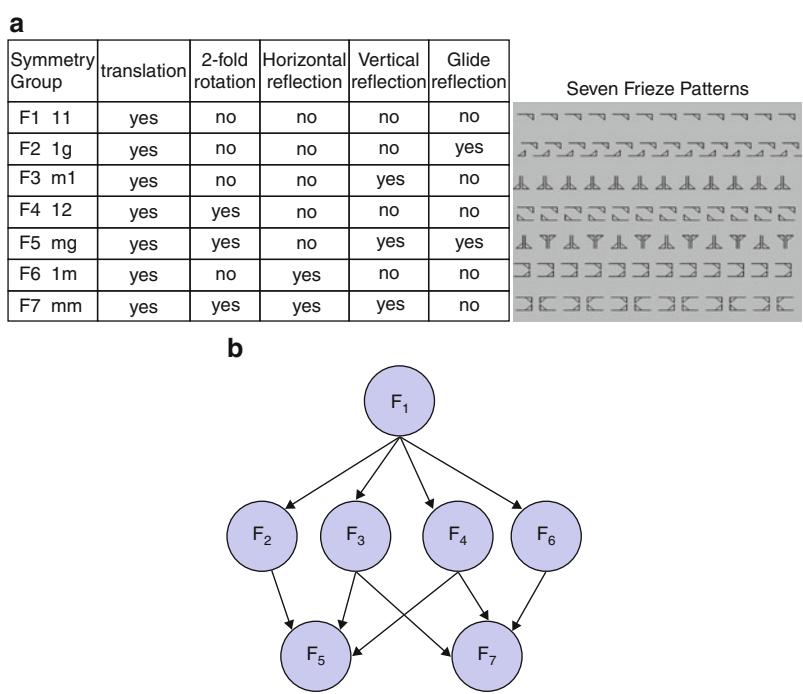
Frieze Symmetry Group

A frieze pattern is a 2D strip in the plane that is periodic along one dimension. Any frieze pattern P is associated with one of the seven unique symmetry groups (Fig. 4). These seven symmetry groups, denoted by crystallographers as $l1$, lg , ml , $l2$, mg , lm , and mm [22], are called the *frieze groups*. Without loss of generality, assuming the direction of translation symmetry of a frieze pattern is horizontal, the frieze pattern can then exhibit five different types of symmetries (Fig. 4a):

1. Horizontal translation
2. Two-fold rotation
3. Horizontal reflection (reflection axis is placed horizontally)
4. Vertical reflection
5. Horizontal glide reflection composed of a half-unit translation followed by a horizontal reflection

The primitive symmetries in each group (the inner structure of a frieze group) and the relationship

Computational Symmetry, Fig. 4 The five types of primitive symmetries (left table of (a)) and the inner (right of (a)) and inter-structures (b) of the seven frieze groups



among the seven frieze groups (inter-structure of frieze groups) are depicted in Fig. 4a, b, respectively. Each frieze pattern is associated with one of the seven possible frieze groups, subject to the combination of these five primitive symmetries in the pattern (Fig. 4a).

Wallpaper Symmetry Group

A wallpaper pattern is a 2D periodic pattern extending along two linearly independent directions [22, 31] (Fig. 5). Any wallpaper pattern is associated with one of the 17 wallpaper groups. Wallpaper group theory [23] states that all translationally symmetric patterns P_r can be generated by a pair of linearly independent, shortest (among all possible) vectors t_1, t_2 applied to a minimum tile. The *orbits* of this pair of translation symmetry generator vectors form a 2D *quadrilateral lattice*, which simultaneously defines all 2D tiles (partitions the space into its smallest generating regions by its translation subgroup), and a topological lattice structure relating all tiles (Fig. 5a).

Motifs of Wallpaper Patterns

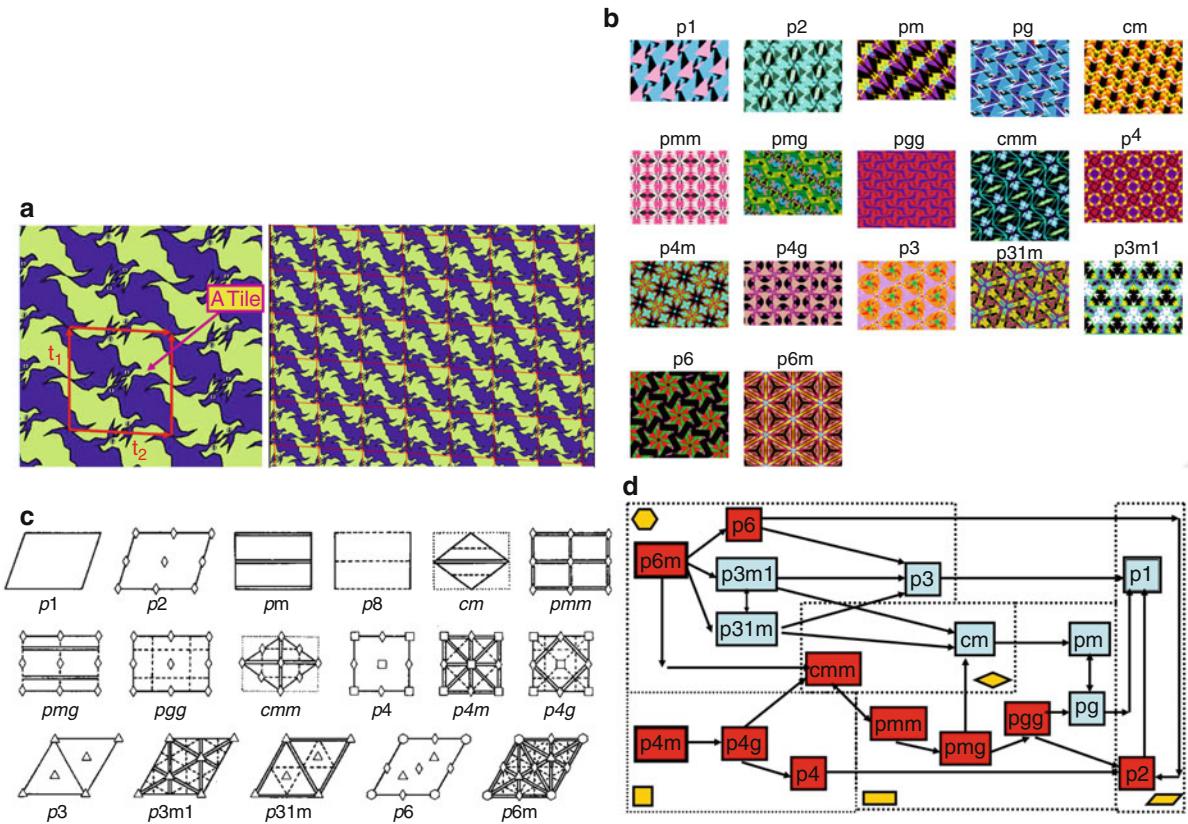
When the translational subgroup of a periodic pattern is determined, it fixes the size, shape, and orientation of the unit lattice, but leaves open the question of where the unit lattice should be anchored in the

pattern. Any parallelogram of the same size and shape carves out an equally good tile that can be used to tile the plane. However, from a perception point of view, some parallelograms produce tiles that are better descriptors of the underlying symmetry of the overall pattern than others. For example, if the whole pattern has some rotation symmetries, a tile located on the rotation centers can represent the global symmetry property of the wallpaper pattern instantly (Fig. 5a). Such *motifs*, as representative tiles of a periodic pattern, can be defined mathematically (and discovered computationally [32, 33]):

Definition 8 A motif of a wallpaper pattern is a tile that is cut out by a lattice whose lattice unit is centered on the fixed point of the largest stabilizer group.

Application

Automatic detection of 2D primitive symmetries (reflection, rotation, translation, glide reflection) from 2D images has been a standing topic in computer vision. The earliest attempt at an algorithmic treatment of bilateral reflection symmetry detection predates computer vision itself [34]. Interested readers can find a long list of published symmetry detection



Computational Symmetry, Fig. 5 (a) A *tile* and a 2D lattice (red) determined simultaneously by the generators t_1, t_2 of the translation subgroup. (b) Sample wallpaper patterns associated with the 17 distinct wallpaper groups. The inner (c) and inter-structures (d) of the 17 wallpaper groups are shown. (c) The symbols $p1, p2, pm, pg, cm, pmm, pmg, pgg, cmm, p4, p4m, p4g, p3, p31m, p3m1, p6$, and $p6m$ are crystallographers’ representations for the wallpaper groups; the diagram is

courtesy of [31]. The *diamond*, *triangle*, *square*, and *hexagon* shapes correspond to two-, three-, four-, and six-fold rotation centers. *Solid single line* and *dotted single line* and *double parallel lines* denote unit translation, glide-reflection, and reflection symmetries, respectively. (d) The subgroup hierarchy, where $A \rightarrow B$ means that B is a subgroup of A (information extracted from [22])

algorithms in [19]. The first quantitative benchmark on reflection, rotation, and translation symmetry detection algorithms can be found in [35, 36].

The first general-purpose, automatic frieze and wallpaper group classification algorithm on real images is published in [32, 33], followed by the classification of *skewed symmetry groups*, in particular wallpaper groups and rotation symmetry groups from affinely and projectively distorted images [37, 38]. More recently, aiming at capturing the translation subgroup of a wallpaper group, several algorithms [39–42] are developed for lattice detection (Fig. 5a) from various locally and/or globally distorted, unsegmented real images.

The applications of computational symmetry are diverse and cross-disciplinary [19]. Some sample

applications in computer vision/graphics include texture replacement in real photos [43], human gait analysis [33, 44], static and dynamic near-regular texture analysis, synthesis, replacement/superimposition and tracking [45–50], image “de-fencing” [51, 52], automatic geotagging [53], fabric categorization [54], architectural facade in-painting [55], multi-view 3D geometry [56], and multi-target tracking [50, 57].

New applications of computational symmetry and group theory attract attentions in urban scene and architectural image analysis, synthesis and 3D-reconstruction [53, 55, 58, 59], multi-target tracking of topology-varying spatial patterns [60], image segmentation and grouping [58, 61–63], transformation estimation [64–66], facial asymmetry as a biometric [67, 68], and an expression and gender cue [69–71].

For biomedical applications, quantified symmetry has also been used to measure symmetry of moving objects such as molecules [72–83], pathological and degenerative volumetric neuroradiology images [84–89], and the firing fields of the grid cells of rats in the context of computational neurosciences [90].

Open Problems

1. Mathematical Symmetries versus Digitized Real-World Data: Well-defined and exhaustively categorized symmetries and symmetry groups become computationally brittle on real-world, noisy, and sometimes distorted digital data.
2. Robust Symmetry Detectors: In spite of years of effort, a robust, widely applicable “symmetry detector” for various types of real-world symmetries remains elusive.
3. Benchmarking: Though an initial effort has been made to quantify and compare the outcome of multiple symmetry detection algorithms [35, 36], the field is yet to see a large-scale, systematic, quantitative evaluation and a publicly available test-image database to gauge the progress in this important, widely applicable research direction.

References

1. Liu Y (2002) Computational symmetry. In: Hargittai I, Laurent T (eds) *Symmetry 2000*. Wenner-Gren international series, vol 80. Portland, London, pp 231–245. ISBN:1 85578 149 2
2. Weyl H (1952) *Symmetry*. Princeton University Press, Princeton
3. Coxeter H, Moser W (1980) Generators and relations for discrete groups, 4th edn. Springer, New York
4. Conway J, Burgiel H, Goodman-Strauss C (2008) *The symmetries of things*. AK Peters, Wellesley
5. Thompson DW (1961) *On growth and form*. Cambridge University Press, Cambridge
6. Giurfa M, Eichmann B, Menzel R (1996) Symmetry perception in an insect. *Nature* 382(6590):458–461
7. Rodríguez I, Gumbert A, Hempel de Ibarra N, Kunze J, Giurfa M (2004) Symmetry is in the eye of the ‘beeholder’: innate preference for bilateral symmetry in flower-naïve bumblebees. *Naturwissenschaften* 91(8):374–377
8. Tyler C (ed) (1996) *Human symmetry perception and its computational analysis*. VSP, Utrecht
9. Blum H (1973) Biological shape and visual science (part i). *J Theor Biol* 38:205–287
10. Nevatia R, Binford T (1977) Description and recognition of curved objects. *Artif Intell* 8(1):77–98
11. Davis L (1977) Understanding shape: angles and sides. *IEEE Trans Comput* 26(3):236–242
12. Blumenthal AF, Davis L, Rosenfeld A (1977) Detecting natural “plateaus” in one-dimensional patterns. *IEEE Trans Comput* 26(2):178–179
13. Kanade T (1981) Recovery of the 3-dimensional shape of an object from a single view. *Artif Intell* 17:75–116
14. Brady M, Asada H (1984) Smoothed local symmetries and their implementation. *Int J Robot Res* 3(3):36–61
15. Biederman I (1985) Human image understanding: recent research and a theory. *Comput Vis Graph Image Process* 32:29–73
16. Pentland A (1986) Perceptual organization and the representation of natural form. *Artif Intell* 28:293–331
17. Lowe D (1985) *Perceptual organization and visual recognition*. Kluwer Academic, Boston
18. Terzopoulos D, Witkin A, Kass M (1987) Symmetry-seeking models and 3D object reconstruction. *Int J Comput Vis* 1:211–221
19. Liu Y, Hel-Or H, Kaplan C, Van Gool L (2010) Computational symmetry in computer vision and computer graphics: a survey. *Found Trends Comput Graph Vis* 5(1/2):1–165
20. Greenberg MJ (1993) *Euclidean and Non-Euclidean geometries: development and history*, 3rd edn. WH Freeman, New York
21. Arnheim R (2004) *Art and visual perception: a psychology of the creative eye*. University of California Press, Berkeley/London
22. Coxeter H (1980) *Introduction to geometry*, 2nd edn. Wiley, New York
23. Grünbaum B, Shephard G (1987) *Tilings and patterns*. WH Freeman, New York
24. Liu Y (1990) Symmetry groups in robotic assembly planning. PhD thesis, University of Massachusetts, Amherst
25. Fedorov E (1885) The elements of the study of figures. (Russian) (2) 21. Zapiski Imperatorskogo S. Peterburgskogo Mineralogichesko Obshchestva (Proc. S. Peterb. Mineral. Soc.), 1–289
26. Fedorov E (1891) Symmetry of finite figures. (Russian) (2) 28. Zapiski Imperatorskogo S. Peterburgskogo Mineralogichesko Obshchestva (Proc. S. Peterb. Mineral. Soc.), 1–146
27. Fedorov E (1891) Symmetry in the plane. (Russian) (2) 28. Zapiski Imperatorskogo S. Peterburgskogo Mineralogichesko Obshchestva (Proc. S. Peterb. Mineral. Soc.), 345–390
28. Bieberbach L (1910) Über die Bewegungsgruppen der n-dimensionalen Euklidischen Räume mit einem endlichen Fundamentalbereich. *Göttinger Nachrichten* 75–84
29. Milnor J (1976) Hilbert’s problem 18. In: *Proceedings of symposia in pure mathematics*, vol 28. American Mathematical Society, Providence. ISBN:0-8218-1428-1 (Browder FE, Mathematical developments arising from Hilbert problems)
30. Henry N, Lonsdale K (eds) (1969) *International tables for X-ray crystallography. Symmetry groups*, vol 1. The Kynoch, England/The International Union of Crystallography, Birmingham
31. Schattschneider D (1978) The plane symmetry groups: their recognition and notation. *Am Math Mon* 85:439–450
32. Liu Y, Collins RT (2000) A computational model for repeated pattern perception using frieze and wallpaper

- groups. In: Computer vision and pattern recognition conference (CVPR'00), Hilton Head, SC. IEEE Computer Society, pp 537–544. http://www.ri.cmu.edu/pubs/pub_3302.html
33. Liu Y, Collins R, Tsin Y (2004) A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Trans Pattern Anal Mach Intell* 26(3):354–371
34. Birkoff G (1932) Aesthetic measure. Harvard University Press, Cambridge
35. Chen P, Hays J, Lee S, Park M, Liu Y (2007) A quantitative evaluation of symmetry detection algorithms. Technical report PSU-CSE-07011 (also listed as technical report CMU-RI-TR-07-36), The Pennsylvania State University, State College, PA
36. Park M, Lee S, Chen P, Kashyap S, Butt A, Liu Y (2008) Performance evaluation of state-of-the-art discrete symmetry detection algorithms. In: IEEE conference on computer vision and pattern recognition (CVPR 2008), Anchorage, pp 1–8
37. Liu Y, Collins RT (2001) Skewed symmetry groups. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR'01), Kauai, HI. IEEE Computer Society, pp 872–879. http://www.ri.cmu.edu/pubs/pub_3815.html
38. Lee S, Liu Y (2010) Skewed rotation symmetry group detection. *IEEE Trans Pattern Anal Mach Intell* 32(9):1659–1672
39. Tuytelaars T, Turina A, Van Gool L (2003) Non-combinatorial detection of regular repetitions under perspective skew. *IEEE Trans Pattern Anal Mach Intell* 25(4):418–432
40. Hays J, Leordeanu M, Efros A, Liu Y (2006) Discovering texture regularity as a higher-order correspondence problem. In: European conference on computer vision (ECCV'06), Graz, Austria
41. Park M, Liu Y, Collins R (2008) Deformed lattice detection via mean-shift belief propagation. In: Proceedings of the 10th European conference on computer vision (ECCV'08), Marseille, France
42. Park M, Brocklehurst K, Collins R, Liu Y (2009) Deformed lattice detection in real-world images using mean-shift belief propagation. *IEEE Trans Pattern Anal Mach Intell* 31(10):1804–1816
43. Tsin Y, Liu Y, Ramesh V (2001) Texture replacement in real images. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR'01), Kauai. IEEE Computer Society, Los Alamitos, pp 539–544
44. Liu Y, Collins R, Tsin Y (2002) Gait sequence analysis using frieze patterns. In: Proceedings of the 7th European conference on computer vision (ECCV'02), a longer version can be found as CMU RI tech report 01–38 (2001), Copenhagen, Denmark
45. Liu Y, Tsin Y (2002) The promise and perils of near-regular texture. In: Texture 2002, Copenhagen, Denmark, in conjunction with European conference on computer vision (ECCV'02), pp 657–671
46. Liu Y, Lin W (2003) Deformable texture: the irregular-regular-irregular cycle. In: Texture 2003, the 3rd international workshop on texture analysis and synthesis, Nice, France, pp 65–70
47. Liu Y, Lin W, Hays J (2004) Near-regular texture analysis and manipulation. *ACM Trans Graph* 23(3):368–376
48. Liu Y, Tsin Y, Lin W (2005) The promise and perils of near-regular texture. *Int J Comput Vis* 62(1-2):145–159
49. Lin W, Liu Y (2006) Tracking dynamic near-regular textures under occlusion and rapid movements. In: Proceedings of the 9th European conference on computer vision (ECCV'06), Graz, Austria, vol 2, pp 44–55
50. Lin W, Liu Y (2007) A lattice-based mrf model for dynamic near-regular texture tracking. *IEEE Trans Pattern Anal Mach Intell* 29(5):777–792
51. Liu Y, Belkina T, Hays H, Lublinerman R (2008) Image defencing. In: IEEE computer vision and pattern recognition (CVPR 2008), Anchorage, pp 1–8
52. Park M, Brocklehurst K, Collins R, Liu Y (2010) Image defencing revisited. In: Asian conference on computer vision (ACCV'10), Queenstown. IEEE Computer Society, pp 1–13
53. Schindler G, Krishnamurthy P, Lublinerman R, Liu Y, Dellaert F (2008) Detecting and matching repeated patterns for automatic geo-tagging in urban environments. In: IEEE computer vision and pattern recognition (CVPR 2008), Anchorage, pp 1–8
54. Han J, McKenna S, Wang R (2008) Regular texture analysis as statistical model selection. In: ECCV08, Marseille
55. Korah T, Rasmussen D (2008) Analysis of building textures for reconstructing partially occluded facades. In: European conference on computer vision (ECCV08), Marseille, pp 359–372
56. Hong W, Yang AY, Ma Y (2004) On symmetry and multiple view geometry: structure, pose and calibration from a single image. *Int J Comput Vis* 60(3):241–265
57. Park M, Liu Y, Collins R (2008) Efficient mean shift belief propagation for vision tracking. In: Proceedings of computer vision and pattern recognition conference (CVPR'08), Anchorage. IEEE Computer Society
58. Park M, Brocklehurst K, Collins R, Liu Y (2010) Translation-symmetry-based perceptual grouping with applications to urban scenes. In: Asian conference on computer vision (ACCV'10), Queenstown. IEEE Computer Society, pp 1–14
59. Wu C, Frahm JM, Pollefeys M (2010) Detecting large repetitive structures with salient boundaries. In: Daniilidis K, Maragos P, Paragios N (eds) European conference on computer vision (ECCV 2010). Lecture notes in computer science, vol 6312. Springer, Berlin/Heidelberg, pp 142–155. doi:10.1007/978-3-642-15552-9_11
60. Liu J, Liu Y (2010) Multi-target tracking of time-varying spatial patterns. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR'10), San Francisco. IEEE Computer Society, pp 1–8
61. Sun Y, Bhanu B (2009) Symmetry integrated region-based image segmentation. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR'08). IEEE Computer Society, Anchorage, Alaska, pp 826–831
62. Yang A, Rao S, Huang K, Hong W, Ma Y (2003) Geometric segmentation of perspective images based on symmetry groups. In: Proceedings of the 10th IEEE international conference on computer vision (ICCV'03), Nice, vol 2, p 1251
63. Levinstein A, Sminchisescu C, Dickinson S (2009) Multiscale symmetric part detection and grouping. In: ICCV, Kyoto

64. Makadia A, Daniilidis K (2006) Rotation recovery from spherical images without correspondences. *IEEE Trans Pattern Anal Mach Intell* 28:1170–1175
65. Tuzel O, Subbarao R, Meer P (2005) Simultaneous multiple 3D motion estimation via mode finding on lie groups. In: Proceedings of the 10th IEEE international conference on computer vision (ICCV'05), Beijing, vol I, pp 18–25
66. Begelfor E, Werman M (2005) How to put probabilities on homographies. *IEEE Trans Pattern Anal Mach Intell* 27:1666–1670
67. Liu Y, Schmidt K, Cohn J, Weaver R (2002) Facial asymmetry quantification for expression invariant human identification. In: International conference on automatic face and gesture recognition (FG'02), Washington, DC
68. Liu Y, Schmidt K, Cohn J, Mitra S (2003) Facial asymmetry quantification for expression invariant human identification. *Comput Vis Image Underst* J 91(1/2):138–159
69. Liu Y, Palmer J (2003) A quantified study of facial asymmetry in 3D faces. In: IEEE international workshop on analysis and modeling of faces and gestures, IEEE, Nice, pp 222–229
70. Mitra S, Liu Y (2004) Local facial asymmetry for expression classification. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR'04). IEEE Computer Society, Washington, DC, pp 889–894. http://www.ri.cmu.edu/pubs/pub_4640.html
71. Mitra S, Lazar N, Liu Y (2007) Understanding the role of facial asymmetry in human face identification. *Stat Comput* 17:57–70
72. Zabrodsky H, Avnir D (1993) Measuring symmetry in structural chemistry. In: Hargittai I (ed) Advanced molecular structure research, vol 1. JAI, Greenwich
73. Zabrodsky H, Avnir D (1995) Continuous symmetry measures, iv: chirality. *J Am Chem Soc* 117:462–473
74. Avnir D, Katzenelson O, Keinan S, Pinsky M, Pinto Y, Salomon Y, Hel-Or H (1997) The measurement of symmetry and chirality: conceptual aspects. In: Rouvray DH (ed) Concepts in chemistry. Research Studies, Somerset, pp 283–324
75. Kanis DR, Wong JS, Marks TJ, Ratner M, Zabrodsky H, Keinan S, Avnir D (1995) Continuous symmetry analysis of hyperpolarizabilities. characterization of second order nonlinear optical response of distorted benzene. *J Phys Chem* 99:11061–11066
76. Yogeve-Einot D, Avnir D (2006) The temperature-dependent optical activity of quartz: from le châtelier to chirality measures. *Tetrahedron Asymmetry* 17:2723–2725
77. Yogeve-Einot D, Avnir D (2004) Pressure and temperature effects on the degree of symmetry and chirality of the molecular building blocks of low quartz. *Acta Crystallogr B60*:163–173
78. Keinan S, Avnir D (2000) Quantitative symmetry in structure-activity correlations: the near $c2$ symmetry of inhibitor/hiv-protease complexes. *J Am Chem Soc* 122:4378–4384
79. Alvarez S, Alemany P, Casanova D, Cirera J, Llunell M, Avnir D (2005) Shape maps and polyhedral interconversion paths in transition metal chemistry. *Coord Chem Rev* 249:1693–1708
80. Pinsky M, Avnir D (1998) Continuous symmetry measures, v: the classical polyhedra. *Inorg Chem* 37:5575–5582
81. Steinberg A, Karni M, Avnir D (2006) Continuous symmetry analysis of NMR chemical shielding anisotropy. *Chem Eur J* 12:8534–8538
82. Pinto Y, Fowler P, Mitchell D, Avnir D (1998) Continuous chirality analysis of model stone-wales rearrangements in fullerenes. *J Phys Chem* 102:5776–5784
83. Keinan S, Avnir D (2001) Continuous symmetry analysis of tetrahedral/planar distortions: copper chlorides and other AB₄ species. *Inorg Chem* 40:318–323
84. Liu Y, Dellaert F (1998) A classification-based similarity metric for 3D image retrieval. In: Proceedings of computer vision and pattern recognition conference (CVPR'98), Santa Barbara. IEEE Computer Society, pp 800–807
85. Liu Y, Collins R, Rothfus W (2001) Robust midsagittal plane extraction from normal and pathological 3D neuro-radiology images. *IEEE Trans Med Imaging* 20(3):175–192
86. Liu Y, Dellaert F, Rothfus W, Moore A, Schneider J, Kanade T (2001) Classification-driven pathological neuroimage retrieval using statistical asymmetry measures. In: International conference on medical imaging computing and computer assisted intervention (MICCAI 2001), Utrecht. Springer, pp 655–665
87. Liu Y, Teverovskiy L, Carmichael O, Kikinis R, Shenton M, Carter C, Stenger V, Davis S, Aizenstein H, Becker J, Lopez O, Meltzer C (2004) Discriminative MR image feature analysis for automatic schizophrenia and alzheimer's disease classification. In: 7th international conference on medical imaging computing and computer assisted intervention (MICCAI 2004), Saint-Malo. Springer, pp 378–385
88. Liu Y, Teverovskiy L, Lopez O, Aizenstein H, Becker J, Meltzer C (2007) Discovery of “biomarkers” for Alzheimer's disease prediction from structural MR images. In: 2002 IEEE international symposium on biomedical imaging: macro to nano, Arlington, pp 1344–1347
89. Teverovskiy L, Becker J, Lopez O, Liu Y (2008) Quantified brain asymmetry for age estimation of normal and AD/MCI subjects. In: 2008 IEEE international symposium on biomedical imaging: nano to macro, Paris, pp 1509–1512
90. Chastain E, Liu Y (2007) Quantified symmetry for entorhinal spatial maps. *Neurocomputing* 70(10–12):1723–1727

Computer Vision

► Rationale for Computational Vision

Computing Architectures for Machine Perception

► High-Performance Computing in Computer Vision

Concept Languages

► Description Logics

Concurrent Mapping and Localization (CML)

- ▶ Exploration: Simultaneous Localization and Mapping (SLAM)

Conditional Random Fields

- ▶ Discriminative Random Fields

Contour Detection

- ▶ Boundary Detection

Convex Minimization

- ▶ Semidefinite Programming

Cook-Torrance BRDF

- ▶ Cook-Torrance Model

Cook-Torrance Model

Abhijeet Ghosh
Institute for Creative Technologies, University of Southern California, Playa Vista, CA, USA

Synonyms

- ▶ Cook-Torrance BRDF

Definition

Cook-Torrance model is an analytic BRDF model that describes the wavelength dependent reflectance property of a surface based on the principles of microfacet theory.

Background

Accurate descriptions of how light reflects off a surface are a fundamental prerequisite for computer vision and graphics applications. Real world materials exhibit characteristic surface reflectance, such as glossy or specular highlights, anisotropy, or off-specular reflection, which need to be modeled for such applications. The surface reflectance of a material is formalized by the notion of the Bidirectional Reflectance Distribution Function (BRDF) [1], which describes the reflected response of a surface in a certain exitant direction to illumination from a certain incident direction over a hemisphere of directions.

Analytical reflection models attempt to describe certain classes of BRDFs using a mathematical representation involving a small number of parameters. The Cook-Torrance model [2] is an analytic isotropic BRDF model that falls under the category of a physics-based model and is based on the microfacet theory of inter-reflection of light at rough surfaces. It extends the Torrance-Sparrow reflectance model, originally developed in the field of applied optics [3], for modeling wavelength dependent effects of reflection. The model predicts both the directional distribution as well as spectral composition of reflected light.

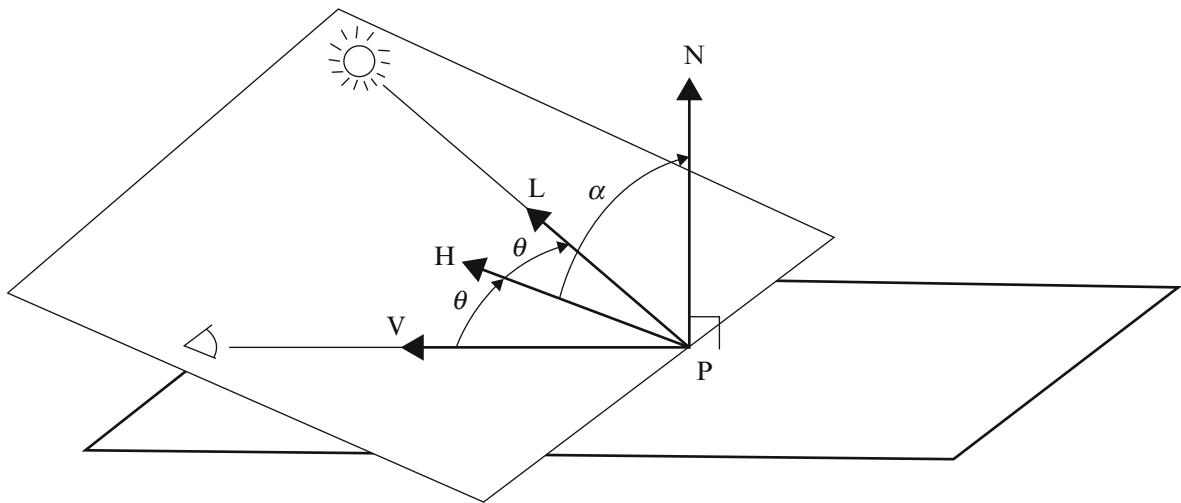
Theory

Given a light source, a surface, and an observer, the Cook-Torrance model describes the intensity and spectral composition of the reflected light reaching the observer. The geometry of reflection is shown in Fig. 1. An observer is looking at a point P on a surface. V is the unit vector in the direction of the viewer, N is the unit normal to the surface, and L is the unit vector in the direction of a specific light source. H is the normalized half-vector between V and L given as

$$H = \frac{L + V}{\|L + V\|}, \quad (1)$$

and is the unit normal of a microfacet that would reflect light specularly from L to V . α is the angle between H and N , and θ is the angle between H and V , so that $\cos \theta = V \cdot H = L \cdot H$.

The main components of the reflectance model are the directionally dependent diffuse and specular



Cook-Torrance Model, Fig. 1 The geometry of reflection [2]

reflection terms, and a directionally independent ambient term. The specular component R_s represents light that is reflected from the surface of the material. The diffuse component R_d originates from internal scattering (in which the incident light penetrates beneath the surface of the material) or from multiple surface reflections (which occur if the surface is sufficiently rough). The directional reflectance is thus given as $R = sR_s + dR_d$, where $s + d = 1$. The model also includes a directionally independent ambient term R_a to approximate the effects of global illumination.

The total intensity of the light reaching the observer is the sum of the reflected intensities from all light sources plus the reflected intensity from any ambient illumination. The basic reflectance model then becomes

$$I_r = I_{ia} R_a + \sum_l (sR_s + dR_d) I_{il} (N \cdot L_l) d\omega_{il}. \quad (2)$$

This formulation accounts for the effect of light sources with different intensities and different projected areas which may illuminate a scene. The next two sub-sections consider the directional and wavelength dependence of the reflectance model.

Directional Distribution

The ambient R_a and diffuse R_d components of the model reflect light independent of the viewing direction. However, the specular component R_s does depend on the viewing direction. The angular spread of

the specular component can be described by assuming that the surface consists of microfacets, each of which reflects specularly [3].

Only facets whose normal is in the direction H contribute to the specular component of reflection from L to V . The specular component is given as

$$R_s = \frac{F}{\pi} \frac{D \cdot G}{(N \cdot L)(N \cdot V)}, \quad (3)$$

where D is the facet slope distribution function, G is the geometric masking and shadowing attenuation term, and F accounts for Fresnel reflectance. The Fresnel term F describes the fraction of light that is reflected from each smooth microfacet. It is a function of incidence angle and wavelength of incident light and is discussed in the next section.

Microfacet Distribution

The facet slope distribution function D represents the fraction of the facets that are oriented in the direction H . Various facet slope distribution functions have been considered by Blinn [4] including the Gaussian model:

$$D = ce^{-(\alpha/m)^2}, \quad (4)$$

where c is the normalization constant.

Beckmann described a model that originated from the study of scattering of radar waves from rough surfaces [5], and is applicable to a wide range of surface conditions ranging from smooth to very rough.

For rough surfaces, the Beckmann distribution is

$$D = \frac{1}{m^2 \cos^4 \alpha} e^{-[(\tan \alpha)/m]^2} \quad (5)$$

In both the facet slope distribution functions, the spread of the specular component depends on the root mean square (rms) slope m . Small values of m signify gentle facet slopes and give a distribution that is highly directional around the specular direction, while large values of m imply steep facet slopes and give a distribution that is spread out with off-specular peak modeled by the Beckmann distribution (see Fig. 2).

For general surfaces with two or more scales of surface roughness, the slope m can be modeled by using a convex weighted combination of two or more distribution functions [6]:

$$D = \sum_j w_j D(m_j), \quad (6)$$

where m_j is the rms slope and w_j the weight of the j th distribution respectively.

Geometric Attenuation

The geometrical attenuation factor G accounts for the shadowing and masking of one facet by another and is discussed in detail in [3, 4]. The following expression is derived for G for microfacets in the shape of v-shaped grooves (see Fig. 3):

$$G = \min \left\{ 1, \frac{2(N \cdot H)(N \cdot V)}{(V \cdot H)}, \frac{2(N \cdot H)(N \cdot L)}{(V \cdot H)} \right\}. \quad (7)$$

Spectral Composition

The ambient, diffuse, and specular reflectances all depend on wavelength. R_a , R_d , and the F term of R_s may be obtained from the appropriate reflectance spectra for the material. Reflectance spectra of thousands of materials can be found in the literature [7–10]. The reflectance data are usually for illumination at normal incidence. These values are normally measured for polished surfaces and must be multiplied by $1/\pi$ to obtain the bidirectional reflectance for a rough surface. Most materials are measured at only a few wavelengths in the visible range (typically around 10–15), so that values for intermediate wavelengths must be interpolated.

The spectral energy distribution of the reflected light is found by multiplying the spectral energy distribution of the incident light by the reflectance spectrum of the surface. An example of this is shown in Fig. 4. The spectral energy distributions of the sun and a number of CIE standard illuminants are available in [11].

Fresnel Reflectance

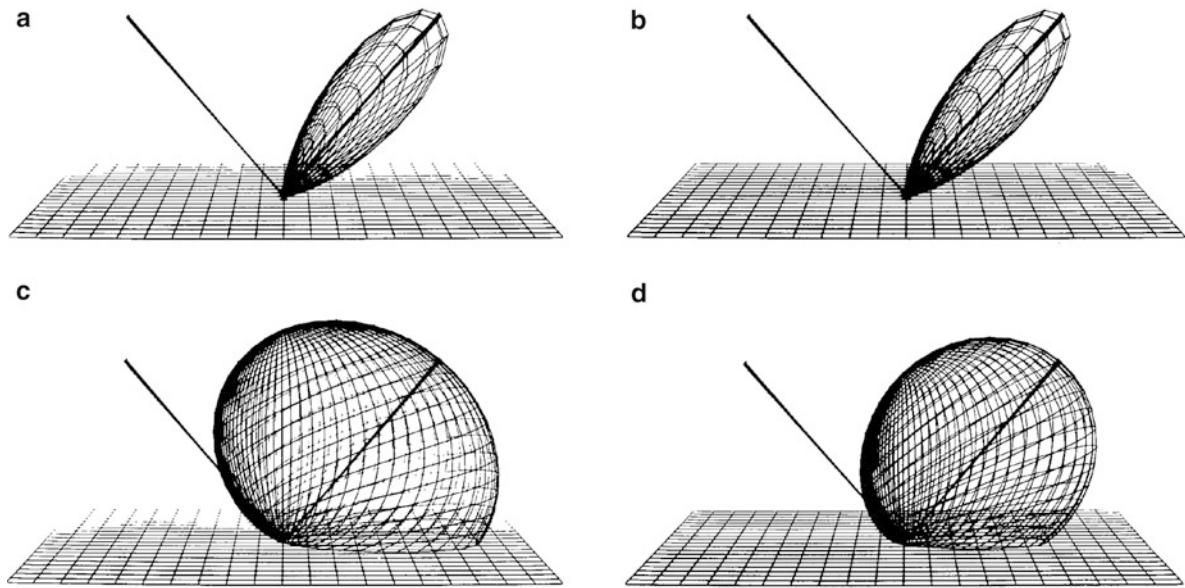
The reflectance F may be obtained theoretically from the Fresnel equation [12]. This equation expresses the reflectance of a perfectly smooth, mirrorlike surface in terms of the index of refraction n (for both metals and dielectrics) and the extinction coefficient k (for metals only) of the surface and the angle of incidence of illumination θ . The Fresnel equation for unpolarized incident light and a dielectric material ($k = 0$) is

$$F = \frac{1}{2} \frac{(g - c)^2}{(g + c)^2} \left\{ 1 + \frac{[c(g + c) - 1]^2}{[c(g - c) + 1]^2} \right\}, \quad (8)$$

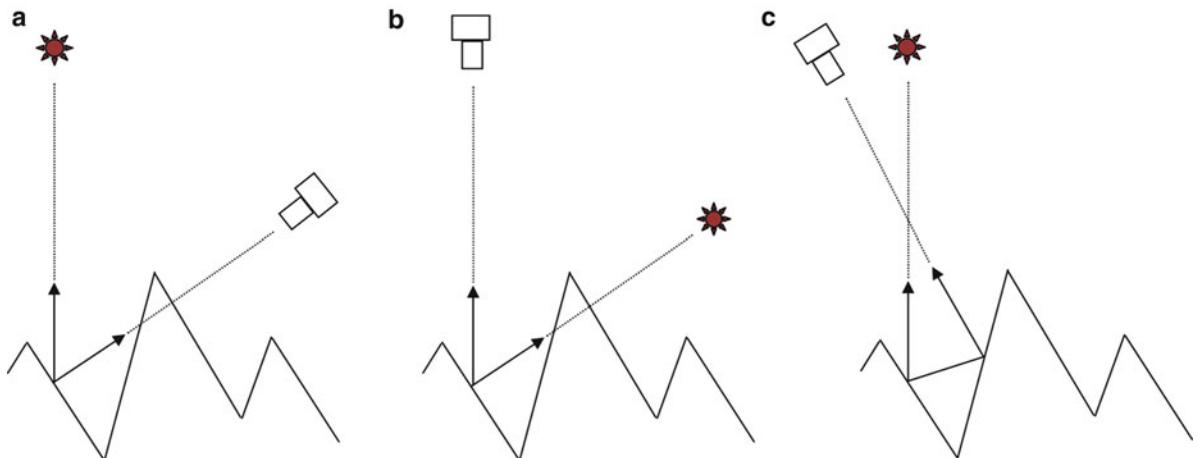
where $c = \cos \theta = V \cdot H$ and $g^2 = n^2 + c^2 - 1$. The dependence of the reflectance on wavelength and the angle of incidence implies that the color of the reflected light changes with the incidence angle, from color of the material at normal incidence to color of the illuminant at grazing incidence.

In general, both n and k vary with wavelength, but their values are frequently not known. On the other hand, experimentally measured values of the reflectance at normal incidence are frequently known. To obtain the spectral and angular variation of F , the following practical approach is adopted: If n and k are known, the Fresnel equation is used. If not, but the normal reflectance is known, the Fresnel equation is fit to the measured normal reflectance for a polished surface. For nonmetals or dielectrics, for which $k = 0$, this immediately gives an estimate of the index of refraction n . For metals, for which k is generally not 0, k is set to 0 to get an effective value for n from the normal reflectance. The angular dependence of F is then available from the Fresnel equation. The above procedure yields the correct value of F for normal incidence and a good estimate of its angular dependence, which is only weakly dependent on the extinction coefficient k .

To illustrate this procedure, consider a dielectric material ($k = 0$) at normal incidence. $\theta = 0$, so $c = 1$,



Cook-Torrance Model, Fig. 2 (a) Beckmann distribution for $m = 0.2$, (b) Gaussian distribution for $m = 0.2$, (c) Beckmann distribution for $m = 0.6$, (d) Gaussian distribution for $m = 0.6$ [2]



Cook-Torrance Model, Fig. 3 Geometric attenuation due to microfacets. (a) Masking. (b) Shadowing. (c) Inter-reflection

$g = n$ and Eq. 8 reduces to

$$F_0 = \frac{(n-1)^2}{(n+1)^2}. \quad (9)$$

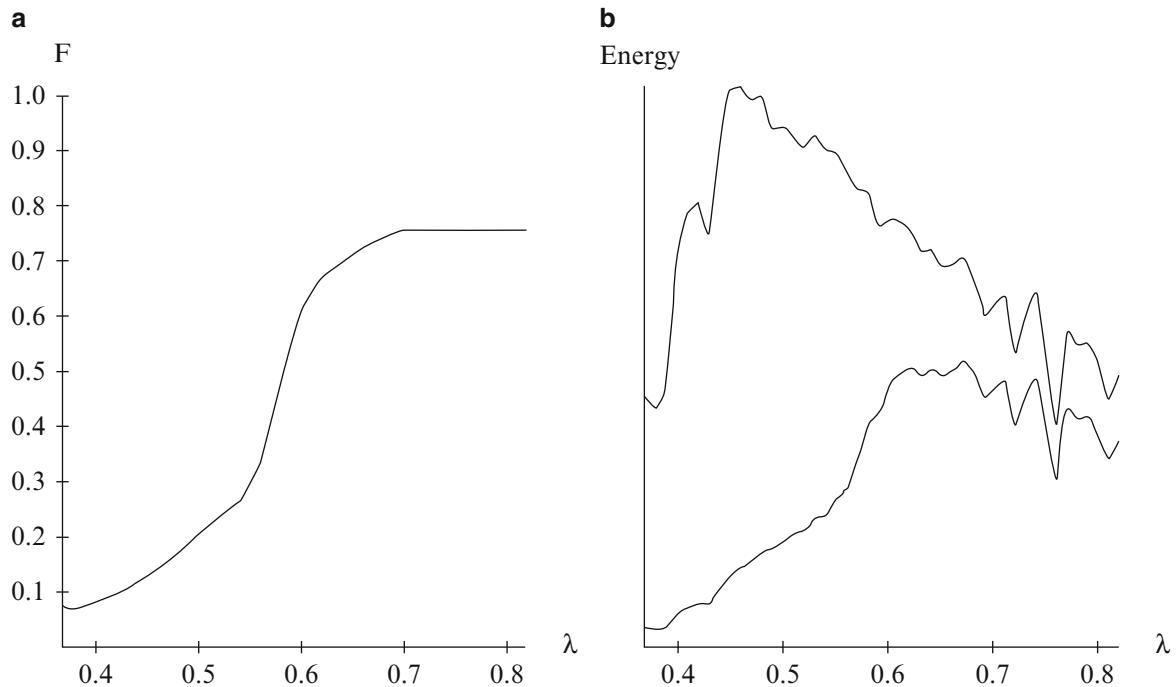
Solving for n gives the equation

$$n = \frac{1 + \sqrt{F_0}}{1 - \sqrt{F_0}}. \quad (10)$$

Values of n determined in this way can then be substituted into the original Fresnel equation to obtain the reflectance F at other angles of incidence. The procedure may then be repeated at other wavelengths to obtain the spectral and directional dependence of the reflectance.

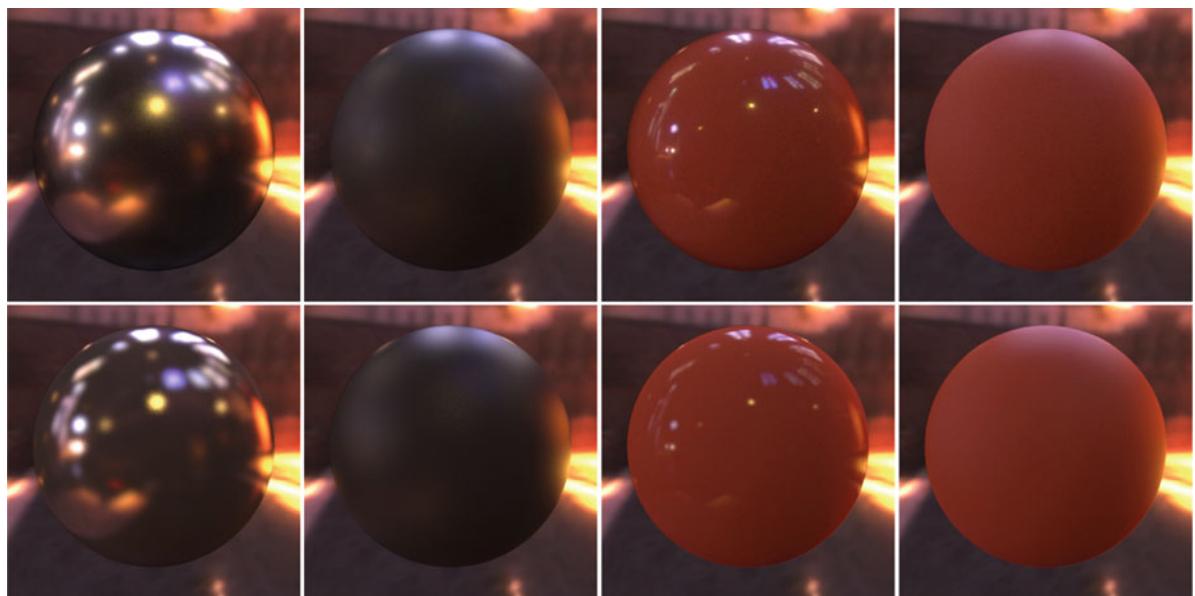
RGB Values

The laws of trichromatic color reproduction are finally used to convert the spectral energy distribution of the



Cook-Torrance Model, Fig. 4 (a) Reflectance of a copper mirror for normal incidence. Wavelength is in micrometers (b) Top curve: Spectral energy distribution of CIE standard illuminant

D6500. Bottom curve: Spectral energy distribution of light reflected from a copper mirror illuminated by D6500 [2]



Cook-Torrance Model, Fig. 5 Measured isotropic BRDFs [14] (top row) and their Cook-Torrance fits [15] (bottom row). Left to right: Nickel, Oxidized steel, Red plastic (specular), Dark red paint



Cook-Torrance Model, Fig. 6 Photograph-rendering pairs of Torrance-Sparrow BRDF fits for modeling spatially varying specular reflectance on faces. Left-pair: [16]. Right-pair: [17]

reflected light to the appropriate RGB values for a particular display device. Every color sensation can be uniquely described by its location in a three dimensional color space. One such color space is called the XYZ space. A point in this space is specified by three coordinates, the color's XYZ tristimulus values. Each spectral energy distribution is associated with a point in the XYZ color space and thus with tristimulus values. If two spectral energy distributions are associated with the same tristimulus values, they produce the same color sensation and are called metamers. The goal, then, is to find the proportions of RGB that produce a spectral energy distribution that is a metamer of the spectral energy distribution of the reflected light.

These proportions are determined by calculating the XYZ tristimulus values that are associated with the spectral energy distribution of the reflected light [11], and then calculating the RGB values (with a linear transform from XYZ followed by a non-linear transform based on display gamma curve) that produce a spectral energy distribution with these tristimulus values [13].

Application

Being a physics-based reflectance model, the Cook-Torrance model has been widely used in computer vision and graphics to model the appearance of real world materials. It has been shown to well approximate the reflectance of many measured isotropic BRDFs in the MERL database [14] ranging from metals, plastics, rubber and fabrics [15] (see Fig. 5).

It has also been successfully applied in computer vision to the problem of uncalibrated photometric stereo [18], and in computer graphics to model the measured surface reflectance of human skin [16, 17] (see Fig. 6).

References

- Nicodemus FE, Richmond JC, Hsia JJ, Ginsberg IW, Limperis T (1977) Geometric considerations and nomenclature for reflectance. National Bureau of Standards: NBS Monograph, vol 160
- Cook R, Torrance KE (1982) A reflection model for computer graphics. ACM Trans Graph 1(1):7–24
- Torrance KE, Sparrow EM (1967) Theory of off-specular reflection from roughened surfaces. J Opt Soc Am 57: 1104–1114
- Blinn JF (1977) Models of light reflection for computer synthesized pictures. Comput Graph 11(2):192–198
- Beckmann P, Spizzichino A (1963) The scattering of electromagnetic waves for rough surfaces. MacMillan, New York
- Porteus JO (1963) Relation between the height distribution of a rough surface and the reflectance at normal incidence. J Opt Soc Am 53(12):1394–1402
- Gubareff GG, Janssen JE, Torborg R (1960) Thermal radiation properties survey: a review of the literature. Honeywell Research Center, Minneapolis
- Purdue University (1970) Thermophysical properties of matter. Vol. 7: thermal radiative properties of metals. Plenum, New York
- Purdue University (1970) Thermophysical properties of matter. Vol. 8: thermal radiative properties of nonmetallic solids. Plenum, New York
- Purdue University (1970) Thermophysical properties of matter. Vol. 9: thermal radiative properties of coatings. Plenum, New York

11. CIE International Commission on Illumination (1970) Official recommendations of the international commission on illumination, Colorimetry (E-1.3.1), CIE 15. Bureau Central de la CIE, Paris
12. Sparrow EM, Cess RD (1978) Radiation heat transfer. McGraw-Hill, New York
13. Meyer G, Greenberg D (1980) Perceptual color spaces for computer graphics. Computer Graph 14(3):254–261
14. Matusik W, Pfister H, Brand M, McMillan L (2003) A data-driven reflectance model. ACM Trans Graph 22(3): 759–769
15. Ngan A, Durand F, Matusik W (2005) Experimental analysis of BRDF models. In: Proceeding the Sixteenth Eurographics Conf on Rendering Techniques, Aire-la-Ville, Switzerland, pp 117–126
16. Weyrich T, Matusik W, Pfister H, Bickel B, Donner C, Tu C, McAndless J, Lee J, Ngan A, Jensen HW, Gross M (2006) Analysis of human faces using a measurement-based skin reflectance model. ACM Trans Graph 25(3): 1013–1024
17. Ghosh A, Hawkins T, Peers P, Frederiksen S, Debevec PE (2008) Practical modeling and acquisition of layered facial reflectance. ACM Trans Graph 27(5)
18. Georgiades 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

Coplanarity Constraint

► [Epipolar Constraint](#)

Corner Detection

Michael Maire
California Institute of Technology, Pasadena,
CA, USA

Related Concepts

► [Edge Detection](#)

Definition

Corner detection is the process of locating points in an image whose surrounding local neighborhoods contain edges of different orientations that intersect at those points.

Background

A corner can be viewed as a special type of interest point. Interest points [1] are distinct local image regions with well-defined positions that are robust to various image deformations, such as changes in viewpoint or lighting. Many corner detection algorithms relax the strict requirement of edge intersection and, for example, instead locate centroids of windows containing high edge energy in multiple directions.

Theory

Most corner detection algorithms operate by scoring local image patches [2–6]. Nonmaximum suppression and thresholding steps can then be applied to localize corners by selecting the peak responses and retaining only those deemed sufficiently salient.

An early approach to corner detection scores patches based on their similarity to neighboring patches [7]. Comparing a patch centered on a corner to patches offset by several pixels in any direction should produce a low similarity score as the edges incident at the corner do not align once shifted. In contrast, patches in uniform regions are identical to their neighboring patches. Comparing patches using the sum of squared differences (SSD) yields score:

$$S_{u,v}(x, y) = \sum_{x_i} \sum_{y_i} [I(x_i + u, y_i + v) - I(x_i, y_i)]^2 \quad (1)$$

for comparing the patch in image I centered at pixel (x, y) to the one offset by u in the x -direction and v in the y -direction. The summation is over the pixels (x_i, y_i) belonging to the patch. An optional weighting factor can be used to decrease the importance of the outer patch region with respect to the center. Since patches centered on edge pixels which are not corners exhibit a large SSD when displaced orthogonal to the edge, but no difference when displaced along it, a robust measure of corner strength C takes the minimum over all possible displacements:

$$C(x, y) = \min_{u,v} S_{u,v}(x, y) \quad (2)$$

In the limit of small displacement, the difference in (Eq. 1) can be approximated by image derivatives:

$$\begin{aligned} I(x_i + \Delta x, y_i + \Delta y) - I(x_i, y_i) &= I_x(x_i, y_i)\Delta x \\ &\quad + I_y(x_i, y_i)\Delta y \end{aligned} \quad (3)$$

yielding:

$$S(x, y) = [\Delta x \quad \Delta y] \begin{bmatrix} \sum_{x_i, y_i} (I_x(x_i, y_i))^2 & \sum_{x_i, y_i} I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_{x_i, y_i} I_x(x_i, y_i) I_y(x_i, y_i) & \sum_{x_i, y_i} (I_y(x_i, y_i))^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (4)$$

Define:

$$A(x, y) = [\nabla I \nabla I^T] \Big|_{(x, y)} \quad (5)$$

The eigenvalues λ_1 and λ_2 of $A(x, y)$ describe the behavior of the local neighborhood of point (x, y) . This region is uniform if λ_1 and λ_2 are both small, an edge if exactly one of λ_1, λ_2 is large, and a corner if both are large.

The Harris corner detector [4] translates this observation into a measure of corner strength given by:

$$C(x, y) = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (6)$$

where k is a parameter. Shi and Tomasi [8] instead use:

$$C(x, y) = \min(\lambda_1, \lambda_2) \quad (7)$$

Lindeberg [5] extends the Harris detector to operate across multiple scales and adds automatic scale selection.

Wang and Brady [9] define corners in terms of curvature using the score:

$$C(x, y) = \nabla^2 I - k |\nabla I|^2 \quad (8)$$

where k is again a constant user-defined parameter (Fig. 1).

Forstner and Gulch [3] take a different approach to corner localization by finding points p whose distance to edges in their local neighborhood is minimal. Specifically,

$$p = \operatorname{argmin}_{(x', y')} \sum_{(x_i, y_i)} D((x_i, y_i), (x', y')) \quad (9)$$

where D is the distance from point (x', y') to the line passing through (x_i, y_i) with orientation orthogonal to the gradient at (x_i, y_i) . The distance is further

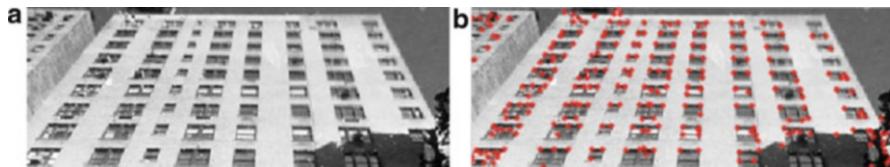
weighted by the gradient magnitude. A similar technique can be used to locate junctions with respect to discrete contour fragments [10].

Application

Interest points, including corners, have found use in a variety of computer vision applications such as image matching, object detection and tracking, and 3D reconstruction. The ability to localize interest points in different views of the same object makes matching and tracking feasible. By design, corner and other interest point detectors respond at locations with rich structure in the surrounding image neighborhood. Thus, these positions are natural choices at which to compute informative feature vectors that describe local image content. The ability to associate these descriptors with interest points further facilitates matching and detection algorithms. In general, the choice of feature descriptor to compute at corners can be application specific and may or may not be coupled with the choice of corner detection algorithm.

While a goal of corner and interest point detectors is to localize the same physical sites across different views, in practice this is only accomplished in a statistical sense. There is usually a significant chance of both missed and spurious detections. Consequently, algorithms built on these components must be robust to errors in detecting individual corners, and instead rely on average detector performance. A typical approach is to utilize a large number of corners or interest points per image for added redundancy.

It is not strictly necessary to use corners or interest points in the process of extracting feature descriptors from images. Alternatives include computing features at sampled edge points or on regions from a segmentation of the image. The sliding



Corner Detection, Fig. 1 Corners detected using the Harris operator followed by nonmaximum suppression and thresholding

window detection paradigm exhaustively scans the image, computing features over all possible image windows.

References

1. Mikolajczyk K, Schmid C (2004) Scale and affine invariant interest point detectors. In: Int J Comput Vis 60(1):63–86
2. Deriche R, Giraudon G (1993) A computational approach for corner and vertex detection. In: Int J Comput Vis 10(2):101–124
3. Forstner W, Gulch E (1987) A fast operator for detection and precise localization of distinct corners. In: ISPPR intercommission conference on fast processing of photogrammetric data, Interlaken, Switzerland pp 281–305
4. Harris C, Stephens M (1988) A combined corner and edge detector. In: Proceedings of the 4th alvey vision conference, University of Manchester, Manchester, UK
5. Lindeberg T (1998) Feature detection with automatic scale selection. In: Int J Comput Vis 30(2):79–116
6. Ruzon MA, Tomasi C (2001) Edge, junction, and corner detection using color distributions. In: IEEE Trans Pattern Analysis and Machine Intelligence, 23(11):1281–1295
7. Moravec H (1980) Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical Report CMU-RI-TR-3, Carnegie-Mellon University
8. Shi J, Tomasi C (1994) Good features to track. In: International conference on computer vision pattern recognition (CVPR), Seattle, WA, pp 593–600
9. Wang H, Brady M (1995) Real-time corner detection algorithm for motion estimation. Image Vis Comput 13:695–703
10. Maire M, Arbeláez P, Fowlkes C, Malik J (2008) Using contours to detect and localize junctions in natural images. In: IEEE conference on computer vision pattern recognition, Anchorage, AK

Cross Entropy

Ying Nian Wu
Department of Statistics, UCLA, Los Angeles,
CA, USA

Definition

Cross entropy is a concept in information theory to measure the independence of two probability distributions.

Theory

For two distributions $p(x)$ and $q(x)$ defined on the same space, the cross entropy is defined as

$$\begin{aligned} H(p, q) &= E_p[-\log q(X)] \\ &= E_p[-\log p(X)] + E_p[\log(p(X)/q(X))] \\ &= H(p) + KL(p, q), \end{aligned}$$

where $H(p) = E_p[-\log p(X)]$ is the entropy of p and $KL(p, q) = E_p[\log(p(X)/q(X))]$ is the Kullback-Leibler divergence from p to q . The Kullback-Leibler divergence is also called relative entropy.

The cross entropy method is a Monte Carlo method for rare event simulation and stochastic optimization [2].

References

1. Cover TM, Thomas JA (1991) Elements of information theory. Wiley, New York
2. Rubinstein RY (1997) Optimization of computer simulation models with rare events. Eur J Oper Res 99:89–112

Curvature

Takayuki Okatani
Graduate School of Information Sciences, Tohoku
University, Sendai-shi, Japan

Synonyms

Curvedness

Related Concepts

► [Geodesics, Distance Maps, and Curve Evolution](#)



Definition

Curvature is a fundamental concept of differential geometry that represents local “curvedness” of some object such as a curve, a surface, and a Riemannian space.

Background

Dealing with the shape of an object is a fundamental issue of computer vision. It is necessary, for example, to represent the two-dimensional or three-dimensional shape of an object, to extract the object shape from various types of images, and to measure similarity between two object shapes. The application of differential geometry to these problems is getting more and more common in modern computer vision. Curvature is one of the most fundamental concept of differential geometry, and its use can be seen throughout all sorts of related problems. This entry explains basic definitions of the curvature of a plane curve and a surface in Euclidean space and summarizes their applications to a few major applications. See [8] for the definition of curvatures of a Riemannian space.

Theory and Application

A plane curve C can be specified by the coordinates $(x(s), y(s))$ of each point, where s is a monotonic function of the arc length. Algebraically, the curvature $\kappa(s)$ of C at a point $(x(s), y(s))$ is defined as

$$\kappa(s) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}, \quad (1)$$

where $\dot{x} = dx/ds$, $\ddot{x} = d^2x/ds^2$, etc. Note that the denominator becomes 1 when s is the arc length itself. Let $\phi(s)$ be the orientation of the tangent to C at $(x(s), y(s))$, that is, $\tan\phi(s) = \dot{y}/\dot{x}$. If s is the arc length, the curvature can be represented as $\kappa = \dot{\phi}(s)$.

Geometrically, the absolute value of κ is equal to the reciprocal of the radius r of the circle osculating the curve C at the point, that is, $|\kappa| = 1/r$, as shown in Fig. 1a; r is called the radius of curvature. The curve C can also be locally approximated by

a second-order polynomial curve. Consider the local coordinates $X Y$ defined by the tangent and the normal vectors to C , as shown in Fig. 2b. The approximating quadratic curve will be given as $Y = \kappa X^2/2$. Note that there is freedom in the choice of the sign of the curvature; in its definition of Eq. (1), the sign depends on the parametrization of s .

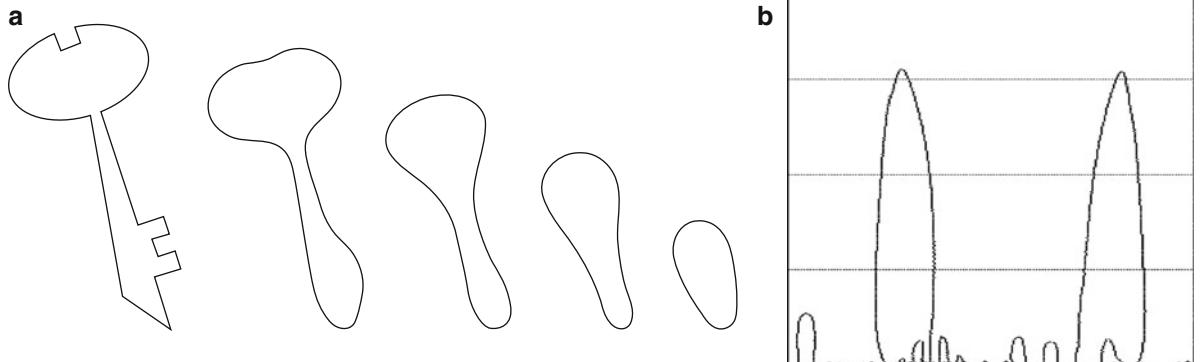
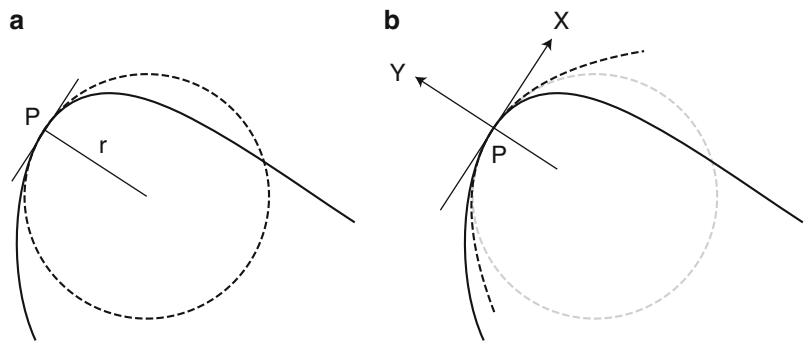
The curvature of a plane curve is effectively used to analyze and/or represent two-dimensional shapes. In [2], the curvature primal sketch is proposed, in which significant changes in curvature along a curve are detected and used for shape representation. In [6], the concept of curvature scale space (CSS) is presented, where the zero-crossings of curvature (i.e., the inflection points of the curve) are used to represent the structure of the curve at varying levels of detail. This technique has been applied to various computer vision problems, such as feature extraction, shape retrieval, and object recognition [5].

The CSS technique is summarized as follows. Let $X(s; \sigma)$ and $Y(s; \sigma)$ be the convolutions of $x(s)$ and $y(s)$ with a Gaussian kernel $g(s, \sigma)$, respectively, that is, $X(s; \sigma) = x(s) * g(s, \sigma)$ and $Y(s; \sigma) = y(s) * g(s, \sigma)$. As shown in Fig. 2a, the curve given by $(X(s; \sigma), Y(s; \sigma))$ gradually becomes smoother, as σ increases. Figure 2b shows a $s - \sigma$ plot of the trajectories of the inflection points. The horizontal axis indicates the normalized arc length s that is scaled to the range $[0-1]$.

The curvature of a plane curve serves as a basic measure of its smoothness. A smoother curve has smaller curvature at each point. Curve evolution like the one in CSS is used in many applications besides shape analysis/representation, where this property with curvature plays a central role.

The active contour model (ACM) [9] was developed to detect the contour of an object in an image by moving an elastic curve from a given initial position to nearby the object contour. This is performed by minimizing the sum of two energy terms, an external term modeling the similarity/dissimilarity to the image edges and an internal term modeling the “elasticity” of the curve. This latter term usually includes the arc length as well as the curvature of the curve to obtain a smoother curve that is more desirable in practice. ACM was later reformulated as a problem of finding local geodesics in a space with Riemannian metric computed from the image, which is known as the geodesic active contour model [4]. The same smoothing property associated with (Euclidean) curvature also plays a key role there.

Curvature, Fig. 1 (a) A plane curve (solid line) and an osculating circle (broken line). The curvature κ at P is the reciprocal of the radius r of the circle. (b) The approximating quadratic curve at P (broken line)



Curvature, Fig. 2 (a) The curvature scale space of a shape. (b) $s-\sigma$ plot of the trajectories of the inflection points of the shape

The evolution of a curve can be fully represented by specifying the normal speed of the curve, that is, the evolution speed measured along the normal vector at each point. Consider a closed curve evolving with speed equal to its curvature, where the sign of the curvature is chosen so that a circle would shrink inward. Its generalization to higher-dimensional space is known as the mean curvature flow. This curve evolution has the properties of “smoothing”; the curve evolves so that its high curvature parts are smoothed out in a finite time. It is shown that any closed curve will become a convex curve and then shrink to a point (Grayson’s theorem [7]).

The level set method (LSM) [17] is a numerical framework for computing such curve evolutions (as well as surface/manifold evolution), which has many advantages to previous methods, such as being able to handle topological changes of the curves. In LSM, a curve is represented as a zero-level curve of an auxiliary function ϕ as $\phi(x, y) = 0$. Then, the curve evolution with normal speed F is represented as $\partial\phi/\partial t = -F|\nabla\phi|$, a time-dependent evolution equation of ϕ .

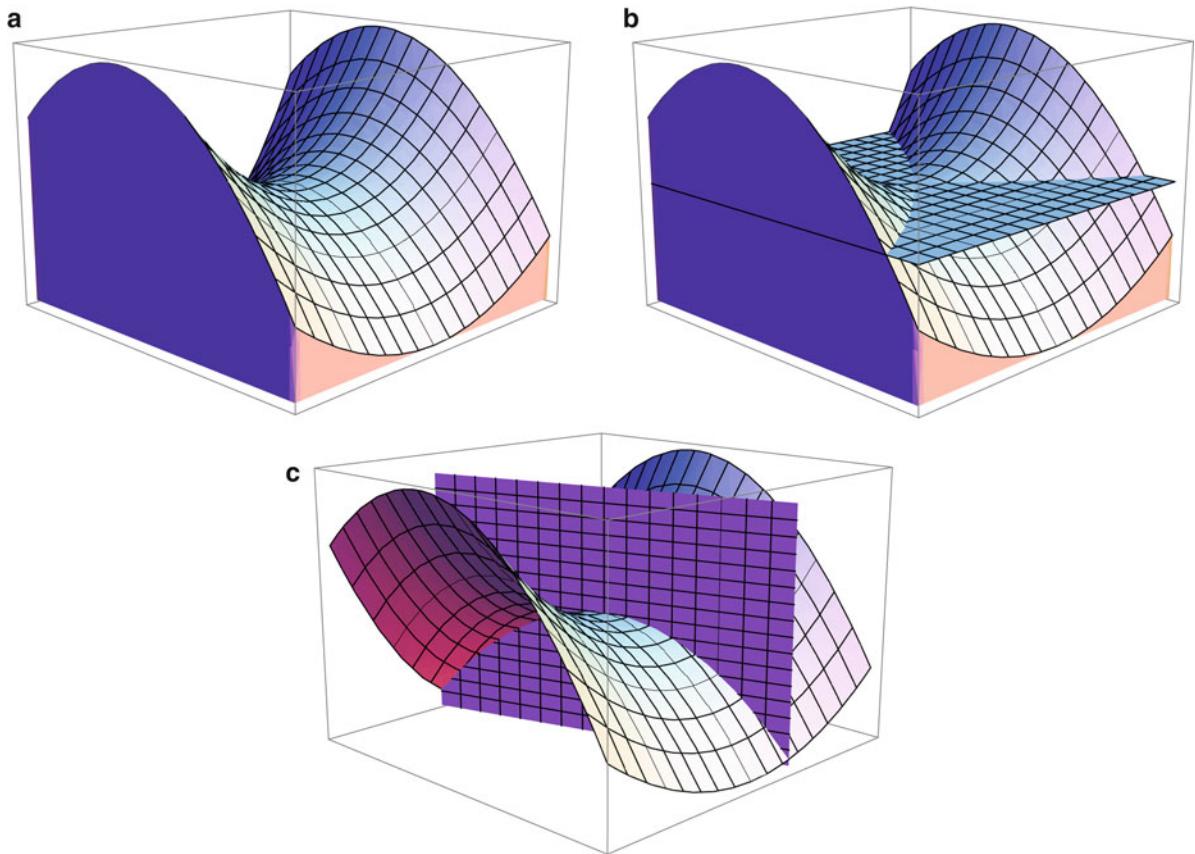
The curvature κ of each point of the level curve is computed using ϕ as

$$\kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \quad (2)$$

where ∇ is the gradient operator $\nabla = (\partial/\partial x, \partial/\partial y)$. Thus, the mean curvature flow is represented as $\partial\phi/\partial t = -\kappa|\nabla\phi|$.

In some problems, the evolution equations of an image $I(x, y)$, which are similar to those of $\phi(x, y)$ above, are considered. An example is the diffusion equation [15] represented as $\partial I/\partial t = \nabla \cdot (c(x, y)\nabla I)$, which has many applications, for example, image denoising/restoration [16] and image inpainting [3]. Although the choice of $c(x, y)$ depends on each application, it often has a term of the curvature of the level curves of $I(x, y)$, which can be also computed by Eq. (2) (When $c(x, y)$ is constant, the resulting image evolution coincides with the Gaussian blurring).

The curvature of a smooth surface S in a three-dimensional space is defined as follows. Consider the



Curvature, Fig. 3 (a) A surface S . (b) The tangent plane to S at a point P in the center. (c) A plane containing the normal vector of S at P . The intersecting curve on the plane gives the normal curvature of S at P

tangent plane to S at a point P of S (Fig. 3b). The normal vector of S at P should be perpendicular to the tangent plane. Then consider a plane containing this normal vector (Fig. 3c). The intersection of the plane with S yields a plane curve on it. The curvature of this plane curve at P , defined in the same manner as above, is called the normal curvature of S at P . The plane has a one-dimensional rotational freedom around the normal vector, and the normal curvature is defined for each of such planes.

Let the maximum and minimum values of the normal curvature at P be k_1 and k_2 . They are called the principal curvatures of S at P . Consider a local coordinate frame XYZ whose origin is located at P and Z axis coincides with the normal vector. The surface is locally approximated by a second-order polynomial surface

$$Z = \frac{1}{2} [X \quad Y] \mathbf{H} \begin{bmatrix} X \\ Y \end{bmatrix}, \quad (3)$$

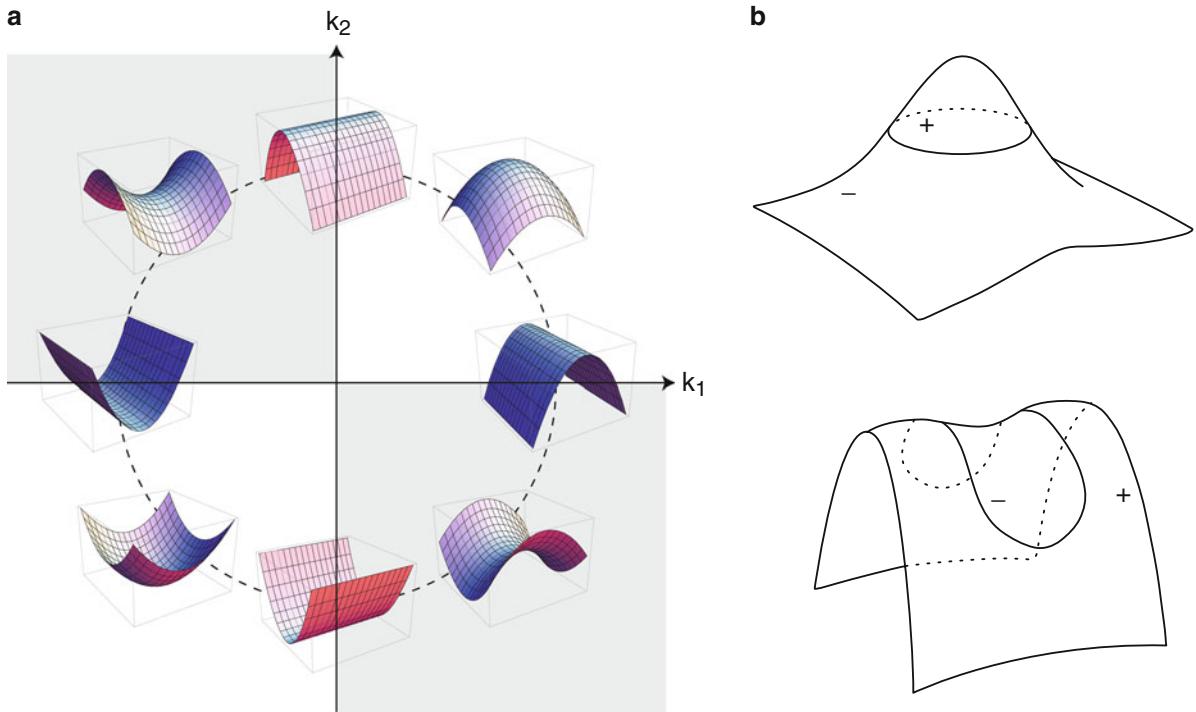
where \mathbf{H} is a 2×2 symmetric matrix. Specifying a direction in the XY plane by a two-dimensional normalized vector \mathbf{v} ($\|\mathbf{v}\|^2 = 1$), the normal curvature for the direction \mathbf{v} is given by $\mathbf{v}^\top \mathbf{H} \mathbf{v}$. The eigenvalues of \mathbf{H} are the same as the principal curvatures k_1 and k_2 , and their associated eigenvectors the corresponding directions \mathbf{v} 's. The Gaussian curvature of S at P is defined as a product of principal curvatures k_1 and k_2 ,

$$\kappa = k_1 k_2, \quad (4)$$

and the mean curvature of S at P is defined as their mean,

$$h = \frac{k_1 + k_2}{2}. \quad (5)$$

Local shapes of a surface are classified by the signs of the principal curvatures k_1 and k_2 , as shown in Fig. 4a [10]. A point at which k_1 and k_2 have the same sign, that is, the Gaussian curvature $\kappa > 0$, is called



Curvature, Fig. 4 (a) Classification of local shapes according to principal curvatures k_1 and k_2 . The first and third quadrants are where the surface point is elliptical, and the second and

fourth quadrants are where it is hyperbolic. (b) Examples of the surface segmentation based on the sign of the Gaussian curvature

an elliptical point. If k_1 and k_2 have different signs, that is, $\kappa < 0$, then the point is called a hyperbolic point. If either k_1 or k_2 vanishes, that is, $\kappa = 0$, then the point is called a parabolic point. Using this classification, a smooth surface may be segmented into finite regions depending on the sign of the curvatures; Fig. 4b shows examples of the surface segmentation based on the Gaussian curvature signs.

This classification method is used in all sorts of applications. For example, it is used in the detection of features such as corners and edges in images, where the images or their variants are regarded as the surface whose local shape is classified; see, for example, [11]. In [18], the shape of an object obtained as range data is represented based on its curvature. It is shown in [1, 13] that similar curvature-based shape representation can be computed from multiple images taken under different illumination directions but without detailed knowledge on the process of the image formation.

More advanced forms of curvatures, such as the curvature of higher-dimensional submanifolds and the Riemannian curvature tensors, are used in recent studies. In [12], several Riemannian metrics on the

space of two-dimensional shapes are studied. In [14], biases of the maximum likelihood estimates derived for the problems of estimating some geometric structure from images (e.g., the epipolar geometry) are related to the curvature of the hypersurfaces given by the geometric structure.

References

- Angelopoulou E, Wolff LB (1998) Sign of gaussian curvature from curve orientation in photometric space. IEEE Trans Pattern Anal Mach Intell 20(10):1056–1066
- Asada H, Brady M (1986) The curvature primal sketch. IEEE Trans Pattern Anal Mach Intell 8(1):2–14
- Bertalmio M, Sapiro G, Caselles V, Ballester C (2000) Image inpainting. In: Proceedings of SIGGRAPH 2000, New Orleans, pp 417–424
- Caselles V, Kimmel R, Sapiro G (1997) Geodesic active contours. Int J Comput Vis 22(1):61–79
- Mokhtarian F, Bober M (2003) Curvature scale space representation: theory, Applications, and MPEG-7 Standardization. Kluwer Academic, Dordrecht
- Mokhtarian F, Mackworth A (1992) A theory of multi-scale, curvature-based shape representation for planar curves. IEEE Trans Pattern Anal Mach Intell 14(8):789–805
- Grayson MA (1987) The heat equation shrinks embedded plane curves to round points. J Differ Geom 26(2):285–314

8. Hazewinkel M (ed) (2002) Encyclopaedia of mathematics. Springer. <http://eom.springer.de/C/c027320.htm>
9. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. *Int J Comput Vis* 1:321–331
10. Koenderink JJ (1990) Solid shape. MIT, Cambridge
11. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
12. Michor PW, Mumford D (2004) Riemannian geometries on spaces of plane curves. *J Eur Math Soc* 8:1–48
13. Okatani T, Deugchi K (1999) Computation of the sign of the gaussian curvature of a surface from multiple unknown illumination images without knowledge of the reflectance property. *Comput Vis Image Underst* 76(2):125–134
14. Okatani T, Deugchi K (2009) On bias correction for geometric parameter estimation in computer vision. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR), Miami, pp 959–966
15. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. *IEEE Trans Pattern Anal Mach Intell* 12(7):629–639
16. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. *Phys D* 60:259–268
17. Sethian JA (1999) Level set methods and fast marching methods. Cambridge University Press, Cambridge
18. Vemuri BC, Mitiche A, Aggarwal JK (1986) Curvature-based representation of objects from range data. *Image Vis Comput* 4(2):107–114

Curvedness

► [Curvature](#)

Curves

► [Curves in Euclidean Three-Space](#)

Curves in Euclidean Three-Space

Jan J. Koenderink

Faculty of EEMSC, Delft University of Technology,
Delft, The Netherlands
The Flemish Academic Centre for Science and the
Arts (VLAC), Brussels, Belgium
Laboratory of Experimental Psychology, University
of Leuven (K.U. Leuven), Leuven, Belgium

Synonyms

[Curves](#); [Space curves](#)

Related Concepts

► [Curvature](#); ► [Differential Invariants](#); ► [Euclidean Geometry](#); ► [Osculating Paraboloids](#)

Definition

Space curves are one-parameter manifolds immersed in Euclidean 3D space $\mathbf{r}(s) \subset \mathbb{E}^3$, where $s \in \mathbb{R}$. One requires differentiability to whatever order is necessary, and $\|\frac{d\mathbf{r}(s)}{ds}\|^2 \neq 0$. It is convenient to require $\|\frac{d\mathbf{r}(s)}{ds}\|^2 = 1$, though this can always be achieved through a reparameterization. Such curves are known as “rectified” or “parameterized by arc-length,” and one writes $\ddot{\mathbf{r}}(s)$ for the partial derivative with respect to arc-length ($\mathbf{r}'(t)$ will be used if the parameter t is not arc-length). As discussed below, in addition one requires $\ddot{\mathbf{r}}(s) \neq 0$ for a generic space curve. The very notion of “rectifiable” is of course Euclidean. Curves in non-Euclidean spaces (affine, projective) have to be handled in appropriate ways.

Background

The classical theory of curves starts with Newton and Leibniz; it was brought in the form presented here in the course of the eighteenth and nineteenth century.

Theory

In differential geometry the “shape” of a curve is defined locally as a set of differential invariants that are algebraic combinations of derivatives $\{\mathbf{r}, \dot{\mathbf{r}}, \ddot{\mathbf{r}}, \dots\}$, and that are invariant with respect to Euclidean motions (notice that “congruencies” would assign the same shape to helices of opposite chirality). A complete set of such differential invariants, specified as a function of the parameter s (arc-length), allows one to construct the curve on the basis of this, up to arbitrary motions. Such a specification of the curve is known as its “natural equations.”

In performing coordinate-wise operations one has to refer to a fiducial frame, most conveniently an orthonormal basis. All equations will take on their simplest form in a frame that is especially fit to the curve. The classical Frenet-Serret frame is one way to

achieve this. As one moves along the curve the frame will rotate in various ways. The simplest description expresses the instantaneous motion of the frame in terms of the frame itself. The differential invariants have geometrically intuitive interpretations in such a system. This insight is due to Elie Cartan, though already implicit in the classical formulation.

These are the basic insights of the classical theory. A short derivation with appropriate geometrical interpretations follows.

The first derivative of position $\mathbf{r}(s)$ with respect to arc-length s is (by construction) a unit vector known as the *tangent* to the curve, denoted $\mathbf{t}(s) = \dot{\mathbf{r}}(s)$. Geometrically the tangent is the direction of the curve, it is the limit of the difference of two points $\mathbf{r}(s_1) - \mathbf{r}(s_2)$ (with $s_1 > s_2$) of the curve divided by the chord-length, that is, $\|\mathbf{r}(s_1) - \mathbf{r}(s_2)\|$, as the points approach each other infinitesimally. Thus the expression $\mathbf{r}(s_0) + (s - s_0)\mathbf{t}(s_0)$ is a first order approximation, the tangent line, to the curve.

Because the tangent is a unit vector, one has $\mathbf{t} \cdot \mathbf{t} = 0$. (Note the Euclidean nature of this!) Thus the second-order derivative $\ddot{\mathbf{r}}$ is orthogonal to the tangent. We write $\ddot{\mathbf{r}} = \kappa \mathbf{n}$, where the unit vector $\mathbf{n}(s)$ is the “normal” to the curve, and $\kappa(s)$ the “curvature.” Notice that the normal would be undefined in the case the tangent did not change direction. For a generic space curve we have to require $\kappa > 0$ throughout (though of course the choice of sign is arbitrary). This is different from planar curves for which a signed curvature makes sense. Thus planar curves may have points of inflection, whereas this notion makes no sense for space curves (Fig. 1).

The normal and the tangent define a plane, the so-called osculating plane of the curve. It is the limit of the plane spanned by three points of the curve as the points approach each other infinitesimally. One might say that, at least locally, the curve lies in its osculating plane (remember that “to osculate” means literally “to kiss”). The three points also define a circle (lying in the osculating plane), whose radius can be shown to be $1/\kappa$. (An easy way to show this is to write down the second order Taylor development of the curve.) Thus, locally, the curve is like a circular arc of radius $1/\kappa$ in the osculating plane, moving in the tangent direction. The curvature measures the degree to which the curve veers away from the tangent direction, into the direction of the normal. Notice that $\ddot{\mathbf{r}} \cdot \mathbf{n} = \kappa$, and that the curvature is a scalar that does not depend upon the

coordinate frame. The curvature is the first example of a differential invariant.

In Euclidean space \mathbb{E}^3 the tangent and the normal imply a third vector orthonormal to them both. It is $\mathbf{b} = \mathbf{t} \times \mathbf{n}$, known as the “binormal.” This is again a very Euclidean construction, the vector product being a Euclidean 3D concept. The orthonormal frame $\{\mathbf{t}, \mathbf{n}, \mathbf{b}\}$ is the Frenet-Serret frame. It is tightly connected to the curve and a complete basis of \mathbb{E}^3 . Thus it is perfectly suited to describe the third order derivative $\ddot{\mathbf{r}}$. The obvious move is to express \mathbf{n} in terms of the tangent and the binormal (given the fact that the normal is a unit vector this should be possible). Thus one writes $\mathbf{n} = -\kappa \mathbf{t} + \tau \mathbf{b}$, where τ is another differential invariant known as the torsion of the curve. The reason for the term $-\kappa \mathbf{t}$ is that $\mathbf{t} = \kappa \mathbf{n}$: the frame turns about the binormal with angular speed κ (Fig. 2). The third derivative itself then is $\ddot{\mathbf{r}} = -\kappa^2 \mathbf{t} + \kappa \mathbf{n} + \tau \mathbf{b}$.

The torsion (sometimes called “second curvature”) has a simple geometrical interpretation. It is the angular rate of change of the attitude of the osculating plane.

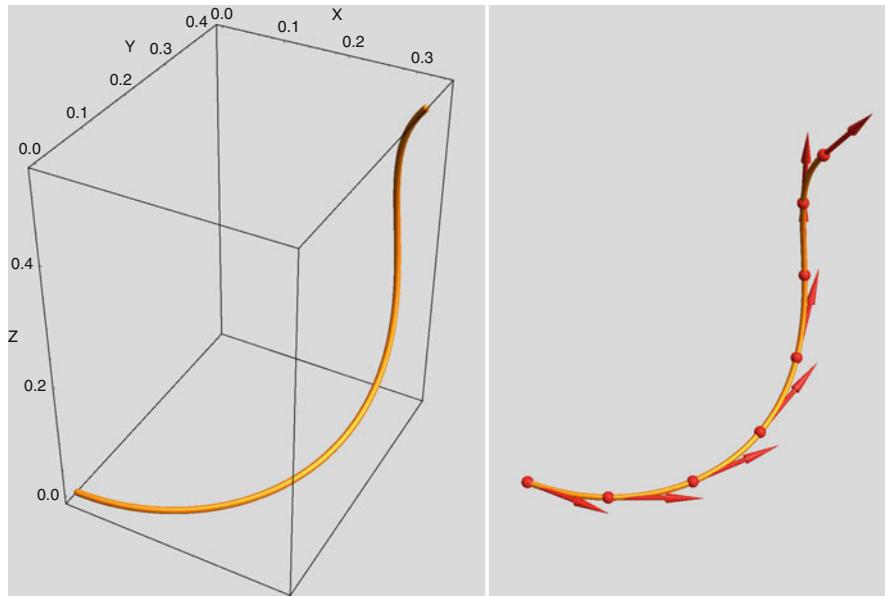
Notice that the metrical structure of \mathbb{E}^3 is used in an essential manner in all constructions thus far. The classical theory of curves cannot be used in spaces with different structures, even in homogeneous spaces such as affine or projective spaces. Of course, a theory of curves can be developed for such spaces too, but the differential invariants, although often denoted “curvature” and “torsion,” will have meanings that are completely distinct from the curvature or torsion of curves in Euclidean space. The reader should be keenly aware of this, as non-Euclidean spaces occur frequently and naturally in computer vision and image processing applications, the best known being the affine and projective 3D spaces, as well as “graph space.”

The structure found thus far can be appreciated from a straightforward Taylor expansion:

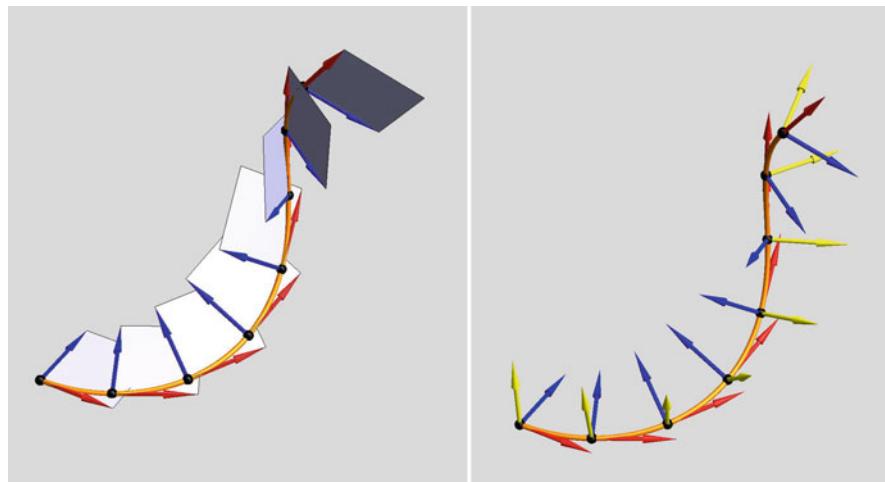
$$\mathbf{r}(s) = \mathbf{r}(0) + \dot{\mathbf{r}}s + \ddot{\mathbf{r}}\frac{s^2}{2!} + \ddot{\mathbf{r}}\frac{s^3}{3!} + O[s]^4, \quad (1)$$

which in terms of the Frenet-Serret frame is (the “canonical representation”):

$$\begin{aligned} \mathbf{r}(0) &+ \left(s - \kappa^2 \frac{s^3}{3!} + \dots \right) \mathbf{t} + \left(\kappa \frac{s^2}{2!} + \kappa \frac{s^3}{3!} + \dots \right) \mathbf{n} \\ &+ \left(\kappa \tau \frac{s^3}{3!} + \dots \right) \mathbf{b}, \end{aligned} \quad (2)$$



Curves in Euclidean Three-Space, Fig. 1 At left generic space curve. It is curved throughout and “winds” in a single sense. This curve was defined via its natural equations (see below),
 $\kappa(s) = 1 + 3s, \tau(s) = 1 + 5s, 0 < s < 1$. At right the field of tangents along the curve

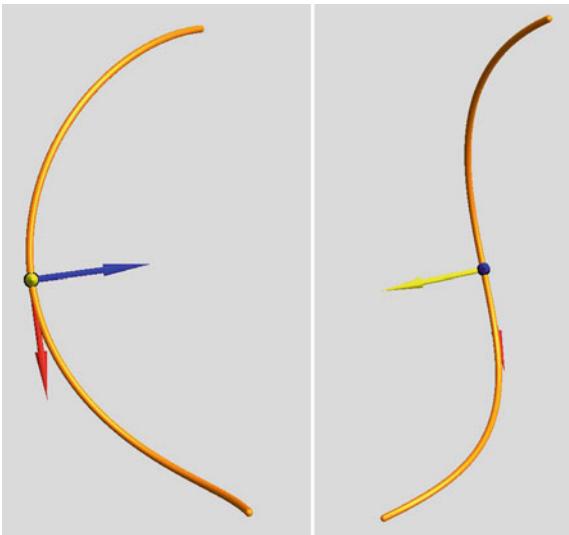


Curves in Euclidean Three-Space, Fig. 2 At left the field of osculating planes along the space curve. Notice how it rotates, revealing the curve to be a “twisted” one. At right the field of

Frenet frames along the curve. Again, notice how it rotates as it moves along the curve

from which the habitus of the curve is readily gleaned: The projection on the osculating plane is (approximately) a parabolic arc, on the normal ($\mathbf{n} \times \mathbf{b}$) plane a cusp, and on the tangential ($\mathbf{b} \times \mathbf{t}$) plane an inflection (Figs. 3 and 4).

Notice that the third order includes a term in the rate of change of curvature, not merely the torsion. The meaning of this becomes evident from another geometrical construction. Four distinct points define a sphere, and in the limit one obtains the osculating sphere at a



Curves in Euclidean Three-Space, Fig. 3 At left a view from the binormal, at right a view from the normal direction. From the binormal direction the curve shows its curvature, from the normal direction one sees an inflection

point of the curve. When the curve is twisted at the point ($\tau \neq 0$), the center of the osculating sphere is given by:

$$\mathbf{c}_{osc} = \mathbf{r}(s) + \frac{1}{\kappa(s)} \mathbf{n}(s) - \frac{\dot{\kappa}(s)}{\kappa(s)^2 \tau(s)} \mathbf{b}(s), \quad (3)$$

and its radius of curvature $\varrho_{osc}(s)$ by:

$$\varrho_{osc} = \sqrt{\varrho^2(s) + \left(\frac{\dot{\varrho}(s)}{\tau(s)} \right)^2}, \quad (4)$$

where ϱ is the radius of the osculating circle. Thus only at “vertices” of the curve ($\dot{\kappa} = 0$) is the osculating circle a great circle of the osculating sphere. The osculating sphere always cuts the osculating plane in the osculating circle though.

The geometrical structure is formulated rather elegantly by the Frenet-Serret formulas (notice the anti-symmetry of the matrix):

$$\begin{pmatrix} \dot{\mathbf{t}} \\ \dot{\mathbf{n}} \\ \dot{\mathbf{b}} \end{pmatrix} = \begin{pmatrix} 0 & +\kappa & 0 \\ -\kappa & 0 & +\tau \\ 0 & -\tau & 0 \end{pmatrix} \begin{pmatrix} \mathbf{t} \\ \mathbf{n} \\ \mathbf{b} \end{pmatrix}. \quad (5)$$

The “natural equations” simply specify $\kappa(s)$ and $\tau(s)$. Using the Frenet-Serret equations one constructs the curve by integration, specifying an arbitrary initial Frenet-Serret frame. Thus the curvature and torsion specify the curve up to arbitrary Euclidean motions.

A useful formalism that extends this is due to Darboux (Fig. 5). The “Darboux vector” is defined as $\mathbf{d} = \tau \mathbf{t} + \kappa \mathbf{b}$. Now one has $\dot{\mathbf{t}} = \mathbf{d} \times \mathbf{t}$, $\dot{\mathbf{n}} = \mathbf{d} \times \mathbf{n}$, and $\dot{\mathbf{b}} = \mathbf{d} \times \mathbf{b}$, thus the Darboux vector is the angular velocity of the “moving trihedron” $\{\mathbf{t}, \mathbf{n}, \mathbf{b}\}$. One immediately concludes that the curvature is the rate of turning about the binormal and the torsion the rate of turning about the tangent. This nicely “explains” the geometrical meaning of the differential invariants κ (the curvature) and τ (the torsion).

The Darboux formalism is by far the simplest to commit to memory. It completely sums up the structure of generic space curves in Euclidean three-space. The Darboux vector also lets one handle degenerate cases easily, for instance that of planar curves (binormal constant), or straight curves (or rather, lines; tangent constant). Finally, the Frenet-Serret equations are written in an easily remembered (because of the cyclic \mathbf{t} - \mathbf{n} - \mathbf{b} -structure) form:

$$\dot{\mathbf{t}} = \mathbf{d} \times (\mathbf{n} \times \mathbf{b}), \quad (6)$$

$$\dot{\mathbf{n}} = \mathbf{d} \times (\mathbf{b} \times \mathbf{t}), \quad (7)$$

$$\dot{\mathbf{b}} = \mathbf{d} \times (\mathbf{t} \times \mathbf{n}). \quad (8)$$

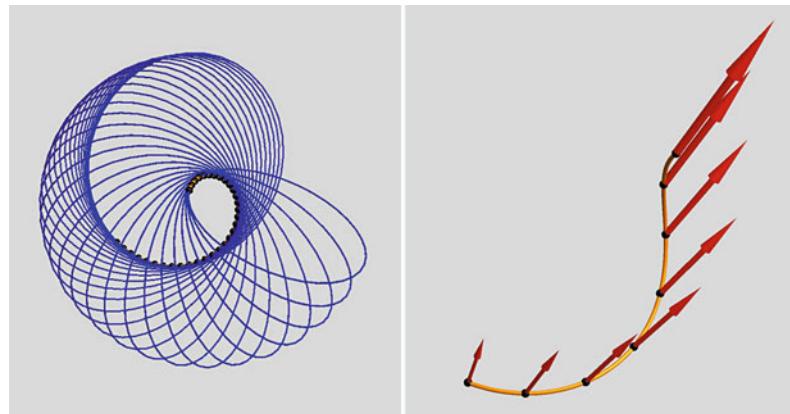
There are a number of geometrical structures related to a curve that are of occasional use. A few of these are discussed below.

So-called spherical images are spherical curves – that are curves on the surface of the unit sphere – related to a curve (Fig. 6). One naturally considers the tangent, normal, and binormal spherical images that are the curves $\mathbf{t}(s)$, $\mathbf{n}(s)$, and $\mathbf{b}(s)$ (notice that these curves are not rectified!). Notice that for straight lines the tangent spherical images degenerate to a point, whereas the other spherical images are undefined. For planar curves the tangent and normal spherical images are degenerated to arcs of great circles, whereas the binormal spherical image degenerates to a point. Special points of the spherical images, like inflections or cusps, relate to special points of the original curve. For some problems of a physical nature it is the spherical image, rather than the curve itself that is of primary

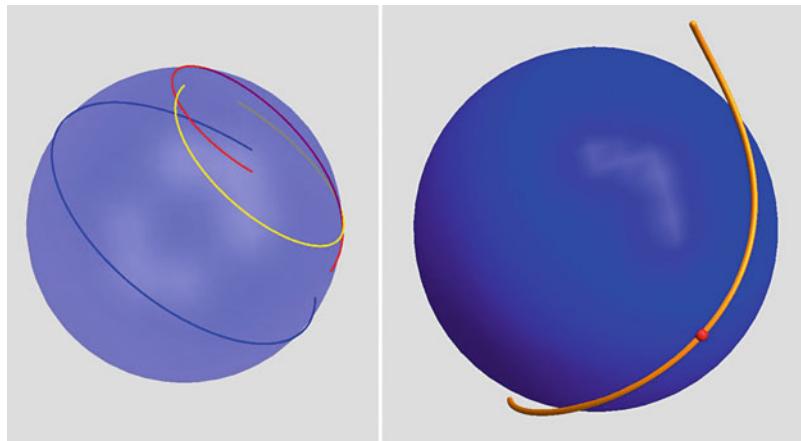


Curves in Euclidean Three-Space, Fig. 4 At center a view from the tangent direction. In this view the curve appears as a cusp. At left and right views from nearby directions

Curves in Euclidean Three-Space, Fig. 5 At left osculating circles along the curve. At right the Darboux vector field along the curve



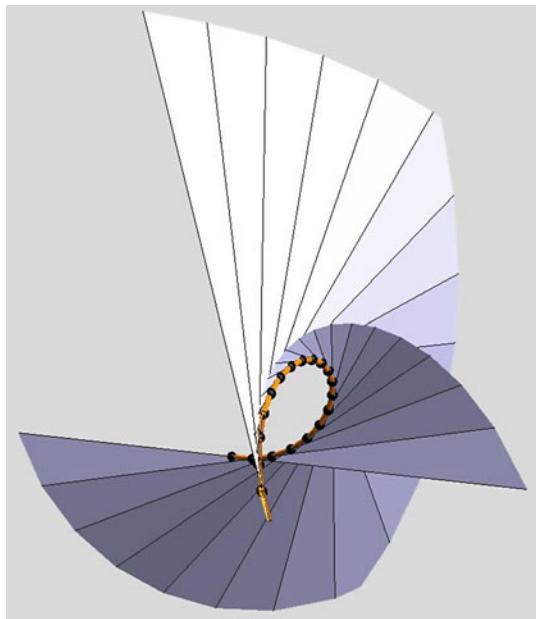
Curves in Euclidean Three-Space, Fig. 6 At left the spherical images associated with the curve (red: tangent image; blue: normal image; yellow: binormal image). At right an osculating sphere at the curve



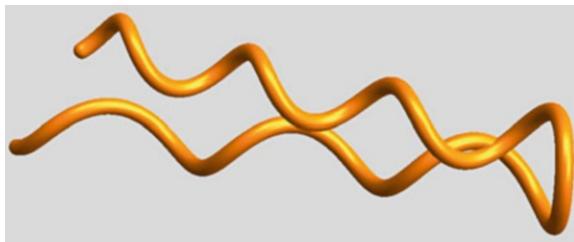
interest. An example is that of specularities on tubular surfaces like human hairs.

The surfaces described by the lines defined by the tangent, normal, binormal, and Darboux vector are also of occasional interest. These are – by construction –

ruled surfaces, though not necessarily developable ones. Their singular features, like the edge of regression in the case of developable surfaces or the line of striction in the case of skew surfaces, have useful geometrical relations to the curve. The best known



Curves in Euclidean Three-Space, Fig. 7 The surface described by the tangents to the curve is a developable surface, the curve being its edge of regression



Curves in Euclidean Three-Space, Fig. 8 Example of a torsion zero. Notice the opposite chirality of the parts of the curves at either side of the torsion zero (which is at the *right* in this picture)

example is the surface of tangent lines, which happens to be developable, with the curve itself as the edge of regression, and the surface of normal lines, which is commonly used to describe surface strips (Fig. 7).

Special points on the curve may also be studied by direct means of course. Perhaps the most obvious

instance is that of a torsion zero (Fig. 8). At a torsion zero the chirality of the curve changes. Whereas the curve generically osculates, but also pierces its osculating plane, the curve merely osculates, but fails to pierce the osculating plane, it being “deflected” by it. Such special points are often introduced by design in telephone cords, and many vines also have frequent torsion zeroes in their tendrils

Additional Problems

This entry describes the Euclidean differential geometry of space curves. In many problems the context is different from Euclidean though. Because the differential invariants introduced here are specific for the Euclidean transformation group, one needs to develop the differential geometry from scratch. Examples frequently occur in computer vision, for instance, and one often works in spaces with mere affine, or even projective structure. Spaces with even less structures are common. A common case involves “isotropic differential geometry” in “graph spaces.”

References

1. Bruce JW, Giblin PJ (1992) Curves and singularities. Cambridge University Press, Cambridge, MA
2. do Carmo MP (1976) Differential geometry of curves and surfaces. Prentice-Hall, Englewood Cliffs
3. Eisenhart LP (2004) A treatise on the differential geometry of curves and surfaces. Dover, New York
4. Gray A (1997) Modern Differential Geometry of Curves and Surfaces with Mathematica, 2nd edn. CRC, Boca Raton
5. Guggenheim H (1977) Differential geometry. Dover, New York
6. Kreyszig E (1991) Differential geometry, Chapter II. Dover Publications, New York
7. Lockwood EH (1961) Book of curves. Cambridge University Press, Cambridge, MA
8. Porteous I (2001) Geometric differentiation. Cambridge University Press, Cambridge, MA
9. Spivak M (1999) A Comprehensive introduction to differential geometry, vol 2. Publish or Perish, Houston