B

Backscattering

▶Retroreflection

Bas-Relief Ambiguity

Manmohan Chandraker
NEC Labs America, Cupertino, CA, USA

Synonyms

Generalized bas-relief (GBR) transformation

Related Concepts

▶Illumination Estimation, Illuminant Estimation; ▶Lambertian Reflectance; ▶Photometric Stereo; ▶Shape from Shadows

Definition

Members of the equivalence class of convex Lambertian surfaces that produce the same set of orthographic images under arbitrary combinations of distant point light sources are related by elements of a three-parameter subgroup of $GL(3)$, called generalized bas-relief (GBR) transformations. This inherent ambiguity in determining the three-dimensional shape of an object from shading and shadow information is called the bas-relief ambiguity.

Background

For a surface $f(x, y)$, the GBR-transformed surface is given by $\bar{f}(x, y) = \mu x + \nu y + \lambda f(x, y)$, where $\mu, \nu \in \mathbb{R}$ and $\lambda \in \mathbb{R}_{++}$. The orthographic image of an object with Lambertian reflectance, illuminated by an arbitrary set of distant point light sources, remains unchanged when the object shape is transformed by a GBR, with an inverse transformation applied on the set of light sources and a corresponding pointwise transformation on the albedos. Further, any continuous transformation that preserves the shading and shadowing configuration for a convex surface must belong to the GBR group [1].

Thus, for a Lambertian surface, any reconstruction or recognition algorithm based on shading and shadow information alone can at best enunciate the shape, albedo, or lighting up to a "bas-relief ambiguity." The ambiguity derives its name from the corresponding low-relief sculpture technique (Italian: *basso rilievo*), which can be understood as a special case of the GBR, where $\lambda < 1$.

Theory

Under orthographic projection, each point on a surface may be represented as $[x, y, f(x, y)]^\top$, where $(x, y) \in \mathbb{R}^2$ is a point on the image plane and f is a piecewise differentiable function. The unit surface normal is given by

$$\hat{\mathbf{n}} = \frac{[-f_x, -f_y, 1]^\top}{\sqrt{f_x^2 + f_y^2 + 1}}. \qquad (1)$$

K. Ikeuchi (ed.), *Computer Vision*, DOI 10.1007/978-0-387-31439-6,

A GBR transformation that maps a surface $f(x, y)$ to $\bar{f}(x, y) = \mu x + \nu y + \lambda f(x, y)$ and the corresponding inverse GBR transformation may be represented as 3×3 linear transformations:

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mu & \nu & \lambda \end{bmatrix}, \quad \mathbf{G}^{-1} = \frac{1}{\lambda}\begin{bmatrix} \lambda & 0 & 0 \\ 0 & \lambda & 0 \\ -\mu & -\nu & 1 \end{bmatrix}. \tag{2}$$

Under the matrix product operation, the set of GBR transformations forms a subgroup of $GL(3)$, the group of 3×3 invertible linear transformations. The unit surface normals of the GBR-transformed surface $\bar{\mathbf{f}}$ are $\frac{\mathbf{G}^{-\top}\hat{\mathbf{n}}}{\|\mathbf{G}^{-\top}\hat{\mathbf{n}}\|}$.

Shadows and Shading

The image formation equation at a point $\mathbf{p} = [x, y, f(x, y)]^\top$ on a Lambertian surface is given by

$$I(x, y) = \mathbf{n}^\top \mathbf{s} \tag{3}$$

where I is the intensity, $\mathbf{n}$ is the product of albedo a and unit surface normal $\hat{\mathbf{n}}$, while $\mathbf{s}$ is the light source direction, scaled by its strength. The point $\mathbf{p}$ lies in an attached shadow if $\mathbf{s}^\top \hat{\mathbf{n}} < 0$, while it lies on a cast shadow boundary if there exists a point $\mathbf{p}'$ on the surface, with unit normal $\hat{\mathbf{n}}'$, such that

$$\mathbf{s}^\top \hat{\mathbf{n}}' = 0, \quad \mathbf{p} - \mathbf{p}' = k\mathbf{s}, \text{ for some } k \in \mathbb{R}_{++}. \tag{4}$$

A point $\mathbf{p} = [x, y, f(x, y)]^\top$ lies in an attached shadow or on a cast shadow boundary in an image produced by the light source $\mathbf{s}$ if and only if the point $\bar{\mathbf{p}} = \mathbf{G}\mathbf{p}$ does so in an image produced by the light source $\bar{\mathbf{s}} = \mathbf{G}\mathbf{s}$, where $\mathbf{G}$ is a GBR transformation given by Eq. (2). Further, the image of a surface $f(x, y)$ with albedo $a(x, y)$, when illuminated by a light source $\mathbf{s}$, is equivalent to the image under the light source $\bar{\mathbf{s}} = \mathbf{G}\mathbf{s}$ of the GBR-transformed surface $\bar{f}$, with a pointwise albedo transformation given by

$$\bar{a} = \frac{a}{\lambda}\left(\frac{(\lambda f_x + \mu)^2 + (\lambda f_y + \mu)^2 + 1}{f_x^2 + f_y^2 + 1}\right)^{\frac{1}{2}}. \tag{5}$$

It follows that the set of images of a Lambertian surface-albedo pair $\{f, a\}$, under all possible combinations of distant light sources, is identical to that of any GBR-transformed surface-albedo pair $\{\bar{f}, \bar{a}\}$ [1] (see Fig. 1). Thus, the illumination cones of surfaces related by a GBR transformation are identical [2].

Existence and Uniqueness

It is shown in [1] that any two convex, smooth surfaces with visible occluding contours that produce the same set of attached shadow boundaries must be related by a GBR transformation. Thus, the GBR transformation is the only one that preserves the set of all images of an object.

While the existence result for the bas-relief ambiguity does not explicitly require convexity of the surface, in practice, the image formation model for concave regions must account for interreflections. It has been shown that modeling diffuse interreflections uniquely determines the shape and lighting [3].

Integrability

In traditional photometric stereo, given images of a point $\mathbf{p}$ under three or more known light sources, one may recover its surface normal $\hat{\mathbf{n}}$ using Eq. (3). However, in uncalibrated photometric stereo where the light sources are unknown, the surface normal and the light sources can be recovered only up to an arbitrary, invertible 3×3 linear transformation, since $\mathbf{n}^* = \mathbf{A}^\top \mathbf{n}$ and $\mathbf{s}^* = \mathbf{A}^{-1}\mathbf{s}$ satisfy Eq. (3) for any $\mathbf{A} \in GL(3)$.

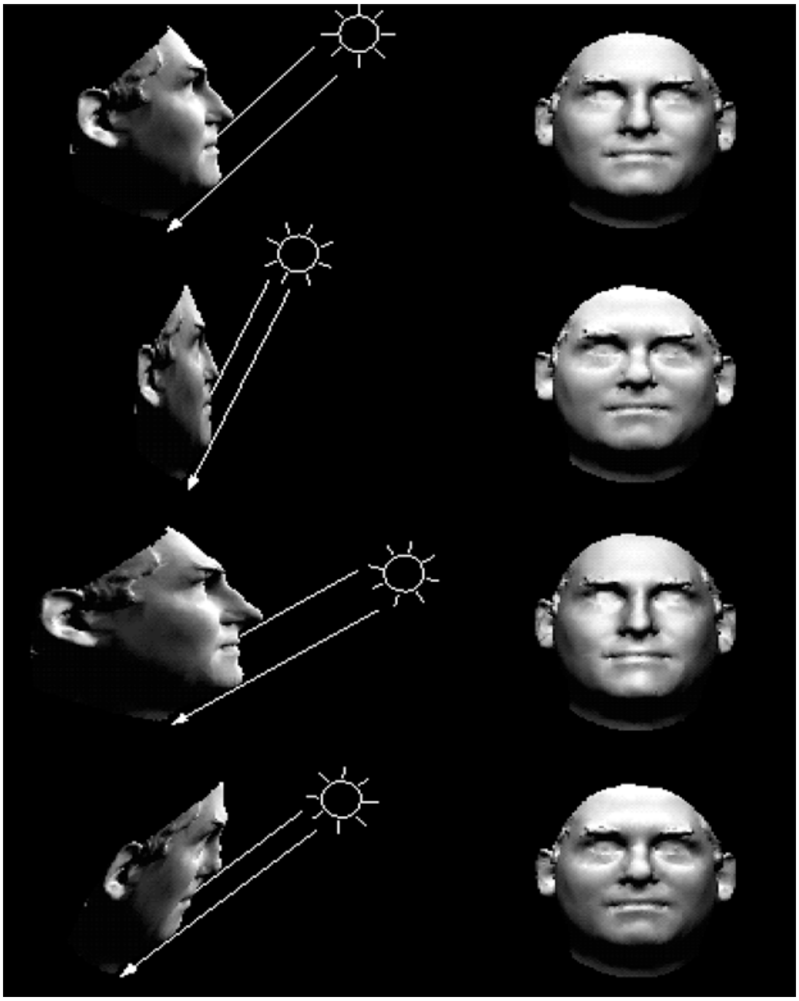
For the recovered normal field to correspond to a surface, it must satisfy the integrability constraint [4]:

$$\frac{\partial}{\partial y}\left(\frac{n_1^*}{n_3^*}\right) = \frac{\partial}{\partial x}\left(\frac{n_2^*}{n_3^*}\right). \tag{6}$$

It is shown in [1] that requiring the recovered normal field to be integrable restricts $\mathbf{A}$ to lie in the group of GBR transformations.

Generalizations

Under perspective projection, the shadows produced by an object under distant or proximal point light sources are the same as those produced by a surface transformed by a generalized perspective bas-relief (GPBR) transformation, with an inverse transformation applied on the light sources [5]. The GPBR is a three-dimensional elation [6] and, in the limiting case



Bas-Relief Ambiguity, Fig. 1 The *left column* shows various GBR transformations applied to a surface, with the corresponding inverse transformation applied to the light source direction. The *top row* is the true shape. The *right column* shows that the shading and shadows produced in an orthographic image of the surface are identical for any GBR transformation (Figure reproduced in part from [1], courtesy of the authors)

of orthographic projection, reduces to the definition of GBR in Eq. (2).

Under orthographic projection from an unknown viewpoint, there exists an ambiguity that corresponds to the group of three-dimensional affine transformations, called the Klein generalized bas-relief (KGBR) ambiguity, such that the set of images of an object is preserved under the action of a KGBR transformation on the shape, lighting, albedos, and viewpoint [7]. In the limiting case of a fixed viewpoint, the KGBR ambiguity reduces to the bas-relief ambiguity.

Application

The bas-relief ambiguity in computer vision explains psychophysical observations of similar unresolved ambiguities in human visual perception [8]. An important consequence of the existence of the bas-relief ambiguity is that any image-based computer vision algorithm, relying on inference based solely on shading and shadow information, can only describe the object up to an arbitrary GBR transformation. Further, it has been established that for any infinitesimal motion of a surface f, there exists a motion for the GBR-transformed surface $\bar{f}$ that produces the same motion field [1]. Thus, an infinitesimal motion does not provide additional cues for disambiguation.

Surface reconstruction up to a GBR transformation can be performed by imposing integrability in uncalibrated photometric stereo [9]. The bas-relief ambiguity may be resolved in practice by incorporating additional information, for instance, priors on albedo distribution [10, 11]. Alternatively, the presence of non-Lambertian effects – such as specular highlights [12], a Torrance-Sparrow reflectance [13], or a spatially unvarying, isotropic, additive non-Lambertian reflectance component [14] – eliminates the GBR ambiguity.

References

1. Belhumeur P, Kriegman D, Yuille A (1999) The bas-relief ambiguity. Int J Comput Vis 35(1):33–44
2. Belhumeur P, Kriegman D (1998) What is the set of images of an object under all possible illumination conditions? Int J Comput Vis 28(3):245–260
3. Chandraker M, Kahl F, Kriegman D (2005) Reflections on the generalized bas-relief ambiguity. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR). IEEE Computer Society, San Diego, CA, pp 788–795
4. Horn B, Brooks M (1986) The variational approach to shape from shading. Comput Vis Graph Image Process 33: 174–208
5. Kriegman D, Belhumeur P (2001) What shadows reveal about object structure. J Opt Soc Am A 18(8):1804–1813
6. Coxeter H (1969) Introduction to geometry, 2nd edn. Wiley, New York
7. Yuille A, Coughlan J, Konishi S (2003) The KGBR viewpoint-lighting ambiguity. J Opt Soc Am A 20(1):24–31
8. Koenderink J, van Doorn A, Christon C, Lappin J (1996) Shape constancy in pictorial relief. Perception 25(2):151–164
9. Yuille A, Snow D (1997) Shape and albedo from multiple images using integrability. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), San Juan, pp 158–164
10. Alldrin N, Mallick S, Kriegman D (2007) Resolving the generalized bas-relief ambiguity by entropy minimization. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), Minneapolis, pp 1–7
11. Hayakawa H (1994) Photometric stereo under a light source with arbitrary motion. J Opt Soc Am A 11(11):3079–3089
12. Drbohlav O, Sara R (2002) Specularities reduce ambiguity of uncalibrated photometric stereo. In: Proceedings of European conference on computer vision (ECCV), Copenhagen, pp 46–60
13. Georghiades A (2003) Incorporating the Torrance-Sparrow model in uncalibrated photometric stereo. In: Proceedings of IEEE conference on computer vision (ICCV), Nice, pp 816–823
14. Tan P, Mallick S, Quan L, Kriegman D, Zickler T (2007) Isotropy, reciprocity and the generalized bas-relief ambiguity. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), Minneapolis, pp 1–8

Behavior Understanding

▶Multi-camera Human Action Recognition

Bidirectional Texture Function and 3D Texture

Kristin Dana
Department of Electrical and Computer Engineering,
Rutgers University, The State University of New
Jersey, Piscataway, NJ, USA

Synonyms

Mesostructure; Microgeometry; Relief texture; Solid texture; Surface roughness; Volumetric texture

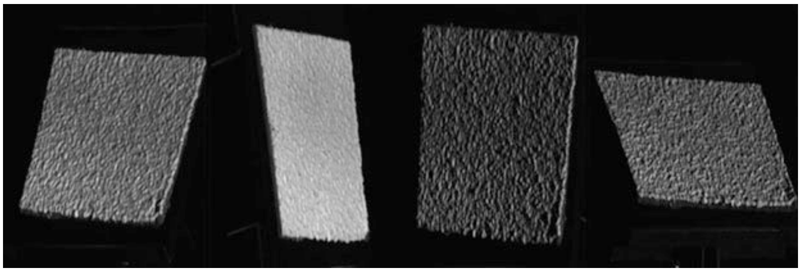
Related Concepts

▶Bidirectional Texture Function and 3D Texture; ▶Bidirectional Feature Histogram; ▶Light Field; ▶Texture Recognition

Definition

In the context of computer vision, texture often refers to a variation of image intensity or color, where the variation exhibits some type of repetition. The terms *2D texture* and *3D texture* provide a more precise definition of texture. A *2D texture* may be a color or shade variation such as a paisley print or zebra stripes. A textured surface can also exhibit geometric variations on the surface such as gravel, grass, or any rough surface. This type of texture is termed *3D texture* [1, 2]. Algorithms developed for 2D texture are generally not useful for 3D texture because appearance varies as a function of viewing and illumination direction.

The difference between 2D and 3D texture is readily apparent when considering photometric effects due to illumination direction and geometric effects due to viewing direction. Consider Fig. 1 with four images of the same surface under different surface tilt angles. The surface geometry does not change, but the illumination and viewing direction is different in each of the images. With a 2D texture model, these changes could be misinterpreted as changes in the texture class. As shown in Fig. 2, the appearance of the texture also changes significantly over the surface of a 3D



Bidirectional Texture Function and 3D Texture, Fig. 1 Four images of the same 3D-textured surface. As the surface tilt and illumination direction varies, surface appearance changes

object. The photometry of 3D texture causes shading and shadowing that vary with illumination direction. The geometry of 3D texture causes a variation in foreshortening and occlusions along the imaged surface. Consider Fig. 3 which illustrates oblique viewing of a 3D-textured surface patch. A similar oblique view of a 2D-textured surface patch gives a uniformly compressed or downsampled version of the frontal view. However, for an obliquely viewed 3D-textured surface patch, there is a non uniform resampling of the frontal view. Consequently, some texture features are compressed in the oblique view, while others expand. Computer graphics algorithms for texture mapping traditionally characterize the texture with a single image. To synthesize oblique views, these texture-mapping algorithms apply a uniform resampling which clearly cannot account for the spatially varying foreshortening and occlusions.

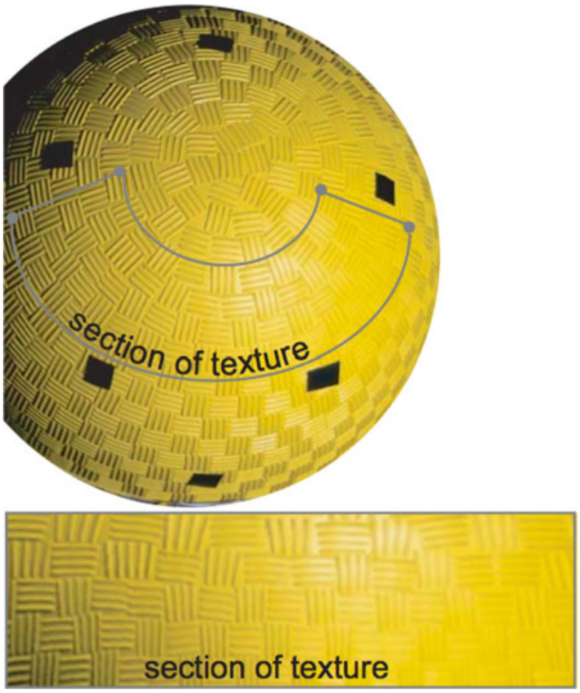
Background

Measurement of 3D texture with a bidirectional texture function (BTF) was introduced in [1, 2]. This work created a database of 3D texture called the CUReT database (Columbia-Utrecht Reflectance and Texture database). This publicly available collection of measurements from real-world surfaces served as a starting point for subsequent work in 3D texture.

Theory

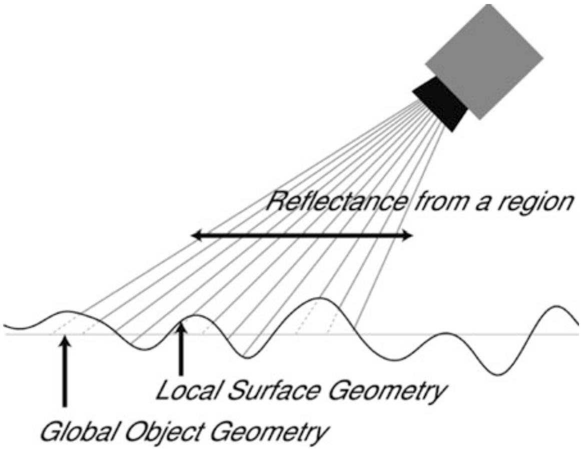
Histogram Models for 3D Texture

Numerous texture models for 2D texture have been developed since the early 1970s and are used in areas like texture mapping, texture synthesis, shape-from-texture, and texture classification and segmentation. These representations include co-occurrence matrices,



Bidirectional Texture Function and 3D Texture, Fig. 2 The local appearance of a 3D-textured object. Notice that local foreshortening, shadowing, and occlusions change across the texture because of the differences in global surface orientation and illumination direction

histograms, power spectra, Gabor filter outputs, textons, wavelets, and Markov random fields. In the late 1990s, the emphasis of texture research expanded to include 3D textures. Analytical models of intensity histograms of 3D-textured surface are developed in [3] and [4]. Intensity histograms are a very basic tool to represent texture but are too simple for most computer vision tasks. A standard framework for texture representations consists of a primitive and a statistical distribution (histogram) of this primitive over space. Intensity is the simplest primitive; however, image features are better primitives to characterize the spatial



Bidirectional Texture Function and 3D Texture, Fig. 3 Geometry of 3D texture. For oblique views of a 3D-textured surface, the sampling rate of the surface depends on the local geometry

relationship of pixels. In order to account for changes with imaging parameters (view/illumination direction), either the primitive or the statistical distribution should be a function of the imaging parameters. Two methods to represent 3D texture are the bidirectional feature histogram (BFH) [5] and the 3D texton method [6]. In the 3D texton method, the primitive is a function of imaging parameters, while in the BFH method, the histogram is a function of imaging parameters. The advantage of the BFH approach is that there is no need for aligning images obtained under different imaging parameters.

Appearance-Based Representations

In computer vision, the precise surface geometry that comprises a 3D texture is often unknown. Instead, images of the 3D texture, i.e., appearance, are used to represent the textured surface. The term bidirectional texture function (BTF) is the appearance of texture as a function of viewing and illumination direction. BTF is typically captured by imaging the surface at a set of possible viewing and illumination directions. Therefore, measurements of BTF are collections of images. The term BTF was first introduced in [2, 7], and similar terms have since been introduced including BSSRDF [8] and SBRDF (spatial BRDF) [9]. SBRDF has a very similar definition to BTF, i.e., BTF is also a spatially varying BRDF.

BTF measurements can be very large because the measurements typically consist of a high-resolution image for every possible viewing and illumination direction. Dense sampling of the illumination and viewing space results in extremely large datasets to represent the surface. For example, if a 3 Mb image is captured for each of the 100 sampled viewing directions and 100 illumination directions, the resulting dataset is 30 Gb. Compact representations of the BTF are clearly important for efficient storage, rendering, and recognition. Methods for compact representations and compression of the BTF are presented in [10–13].

Geometry-Based Representations from Computer Graphics

In computer vision, image-based representations are standard because the surface geometry is typically unknown. However, in computer graphics, the precise geometry of the 3D-textured surface may be known. Rendering 3D textures using a volumetric representation of surface geometry is a common approach. Many of the rendering packages such as OpenGL and Blender use the term 3D texture to refer to volumetric texture. In this definition, the 3D texture is defined by opacity in a volume, instead of the definition here which refers to a surface height variation. In recent work [14], a volumetric representation is used for texture rendering, where the volume is a stack of semitransparent layers obtained using measured BTF data. Historically, volumetric texture methods are also referred to as solid texturing [15–18].

Application

The main applications for 3D texture representations are in recognition, synthesis, and rendering. While many applications use image-based representations, or assume a known geometric model, other applications need to capture the local geometry of 3D texture. Digital archiving of art is an example of such an application where fine-scale surface detail can enhance geometric models. Often the 3D texture is not easily captured with standard laser scanning devices. For example, in capturing the geometry of sculptures, researchers devised ways to capture high-resolution 3D texture such as tool marks [19–21]. An additional method for capturing high-resolution 3D texture geometry uses a specialized texture camera based on the optical properties of curved mirrors [22]. For texture recognition, 3D texture methods have been used in material recognition [6, 23–25]. One of the popular recognition tasks is the recognition of materials from the CUReT database. Another real-world recognition task is the measurement and recognition of skin texture [26–28]. For texture synthesis and rendering, the main problem is to synthesize the appearance of 3D texture. Several authors have developed methods to synthesize and render 3D-textured surfaces using the BTF representation [29–35]. Other authors used an image-based approach to capture and render complex surfaces by direct photography of the full object under various illumination and viewing directions, simultaneously capturing object shape and surface light fields [36–38]. Synthesizing 3D textures via texture morphing enables creating and rendering novel 3D texture [39]. In computer graphics, 3D-textured surfaces were traditionally rendered by bump mapping [40], albeit with limited realism. In more recent years, several new methods have been developed that can efficiently render 3D texture detail. These include view-dependent displacement mapping [41], relief texture mapping [42], the polynomial texture map [43], and a Blender-based rendering method [44].

References

1. Dana KJ, van Ginneken B, Nayar SK, Koenderink JJ (1997) Reflectance and texture of real world surfaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Juan, Puerto Rico, pp 151–157
2. Dana KJ, van Ginneken B, Nayar SK, Koenderink JJ (1999) Reflectance and texture of real world surfaces. ACM Trans Graph 18(1):1–34
3. van Ginneken B, Koenderink JJ, Dana KJ (1999) Texture histograms as a function of irradiation and viewing direction. Int J Comput Vis 31(2–3):169–184
4. Dana KJ, Nayar SK (1998) Histogram model for 3d textures. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Santa Barbara, California, pp 618–624
5. Cula OG, Dana KJ (2004) 3D texture recognition using bidirectional feature histograms. Int J Comput Vis 59(1): 33–60
6. Leung T, Malik J (2001) Representing and recognizing the visual appearance of materials using three-dimensional textons. Int J Comput Vis 43(1):29–44
7. Dana KJ, van Ginneken B, Nayar SK, Koenderink JJ (1996) Reflectance and texture of real-world surfaces. Columbia University Technical Report CUCS-048-96
8. Jensen HW, Marschner SR, Levoy M, Hanrahan P (2001) A practical model for subsurface light transport. In: ACM SIGGRAPH, Los Angeles, California, pp 511–518
9. McAllister DK, Lastra AA, Cloward BP, Heidrich W (2002) The spatial bi-directional reflectance distribution function. In: ACM SIGGRAPH 2002 conference abstracts and applications, SIGGRAPH'02. ACM, New York, pp 265–265
10. Cula OG, Dana KJ (2001) Compact representation of bidirectional texture functions. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Kauai, Hawaii, vol 1. pp 1041–1067
11. Haindl M, Filip J (2007) Extreme compression and modeling of bidirectional texture function. IEEE Trans Pattern Anal Mach Intell 29(10):1859–1865
12. Ruiters R, Klein R (2009) Btf compression via sparse tensor decomposition. Proc EGSR Comput Graph Forum 28(4):1181–1188
13. Wu H, Dorsey J, Rushmeier H (2011) A sparse parametric mixture model for btf compression, editing and rendering. Proc Eurograp Comput Graph Forum 30(2):465–473
14. Magda S, Kriegman D (2006) Reconstruction of volumetric surface textures for real-time rendering. In: Proceedings of the Eurographics symposium on rendering, Nicosia, Cyprus, pp 19–29
15. Perlin K (1985) An image synthesizer. ACM SIGGRAPH 19(3):287–296
16. Kajiya JT, Kay TL (1989) Rendering fur with 3d textures. ACM SIGGRAPH 23(3):271–80
17. Jagnow R, Dorsey J, Rushmeier H (2004) Stereological techniques for solid textures. In: ACM SIGGRAPH 2004 Papers, SIGGRAPH '04. ACM, New York, pp 329–335
18. Kopf J, Fu CW, Cohen-Or D, Deussen O, Lischinski D, Wong TT (2007) Solid texture synthesis from 2d exemplars. In: ACM SIGGRAPH 2007 papers, SIGGRAPH '07. ACM, New York
19. Bernardini F, Martin IM, Rushmeier H (2001) High-quality texture reconstruction from multiple scans. IEEE Trans Vis Comput Graph 7(4):318–332
20. Bernardini F, Martin I, Mittleman J, Rushmeier H, Taubin G (2002) Building a digital model of michelangelo's florentine pieta. IEEE Comput Graph Appl 1(22):59–67
21. Levoy M, Pulli K, Curless B, Rusinkiewicz S, Koller D, Pereira L, Ginzton M, Anderson S, Davis J, Ginsberg J, Shade J, Fulk D (2000) The digital michelangelo project: 3d scanning of large statues. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques. SIGGRAPH '00. ACM/Addison-Wesley, New York, pp 131–144
22. Wang J, Dana KJ (2006) Relief texture from specularities. IEEE Trans Pattern Anal Mach Intell 28(3):446–457
23. Varma M, Zisserman A (2002) Classifying images of materials. In: Proceedings of the European Conference on Computer Vision (ECCV), Copenhagen, Denmark, pp 255–271
24. Cula OG, Dana KJ (2001) Recognition methods for 3d textured surfaces. Proc SPIE Conf Human Vis Electron Imaging VI 4299:209–220
25. Chantler M, Petrou M, Penirsche A, Schmidt M, McGunnigle G (2005) Classifying surface texture while

simultaneously estimating illumination direction. Int J Comput Vis 62(1–2):83–96
26. Cula O, Dana K, Murphy F, Rao B (2004) Bidirectional imaging and modeling of skin texture. IEEE Trans Biomed Eng 51(12):2148–2159
27. Cula OG, Dana KJ, Murphy FP, Rao BK (2005) Skin texture modeling. Int J Comput Vis 62(1/2):97–119
28. 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 25:1013–1024
29. Liu X, Yu Y, Shum H (2001) Synthesizing bidirectional texture functions for real world surfaces. In: ACM SIGGRAPH, Los Angeles, pp 97–106
30. Tong X, Zhang J, Liu L, Wang X, Guo B, Shum H (2002) Synthesis of bidirectional texture functions on arbitrary surfaces. ACM Trans Graph 21(3):665–672
31. Koudelka ML, Magda S, Belhumeur PN, Kriegman DJ (2003) Acquisition, compression and synthesis of bidirectional texture functions. In: 3rd international workshop on texture analysis and synthesis (Texture 2003), Nice, France, pp 59–64
32. Meseth J, Muller G, Klein R (2003) Preserving realism in real- time rendering of bidirectional texture functions. In: Proceedings of the openSG symposium, Darmstadt, Germany, pp 89–96
33. Vasilescu MAO, Terzopoulos D (2004) Tensortextures: multilinear image-based rendering. In: ACM SIGGRAPH 2004 Papers, SIGGRAPH ’04. ACM, New York, pp 336–342
34. Wang J, Tong X, Chen Y, Guo B, Shum H, Snyder J (2005) Capturing and rendering geometry details for btf-mapped surfaces. Vis Comput 21:559–568
35. Kautz J, Boulos S, Durand F (2007) Interactive editing and modeling of bidirectional texture functions. In: ACM SIGGRAPH 2007 papers, SIGGRAPH’07. ACM, New York
36. Matusik W, Pfister H, Ngan A, Beardsley P, Ziegler R, McMillan L (2002) Image-based 3d photography using opacity hulls. In: Proceedings of the 29th annual conference on Computer graphics and interactive techniques. SIGGRAPH’02. ACM, New York, pp 427–437
37. Debevec P, Hawkins T, Tchou C, Duiker H, Sarokin W, Sagar M (2002) Acquiring the reflectance field of a human face. In: ACM SIGGRAPH, San Antonio, pp 145–156
38. Nishino K, Sato Y, Ikeuchi K (2001) Eigentexture method: Appearance compression and synthesis based on a 3d model. IEEE Trans Pattern Anal Mach Intell 23(11): 1257–1265
39. Matusik W, Zwicker M, Durand F (2005) Texture design using a simplicial complex of morphable textures. ACM Trans Graph 25(3):784–794
40. Blinn JF (1978) Simulation of wrinkled surfaces. ACM SIGGRAPH 12(3):286–292
41. Wang L, Wang X, Tong X, Lin S, Hu S, Guo B, Shum H (2003) View-dependent displacement mapping. ACM Trans Graph 22(3):334–339
42. Oliveira MM, Bishop G, McAllister D (2000) Relief texture mapping. In: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH’00. ACM/Addison-Wesley, New York, pp 359–368
43. Malzbender T, Gelb D, Wolters H (2001) Polynomial texture maps. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques, SIGGRAPH’01. ACM, New York, pp 519–528
44. Hatka M, Haindl M (2011) Btf rendering in blender. In: Proceedings of the 10th International Conference on Virtual Reality Continuum and Its Applications in Industry, VRCAI’11. ACM, New York, pp 265–272

Blackbody Radiation

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

Synonyms

Thermal Radiator

Related Concepts

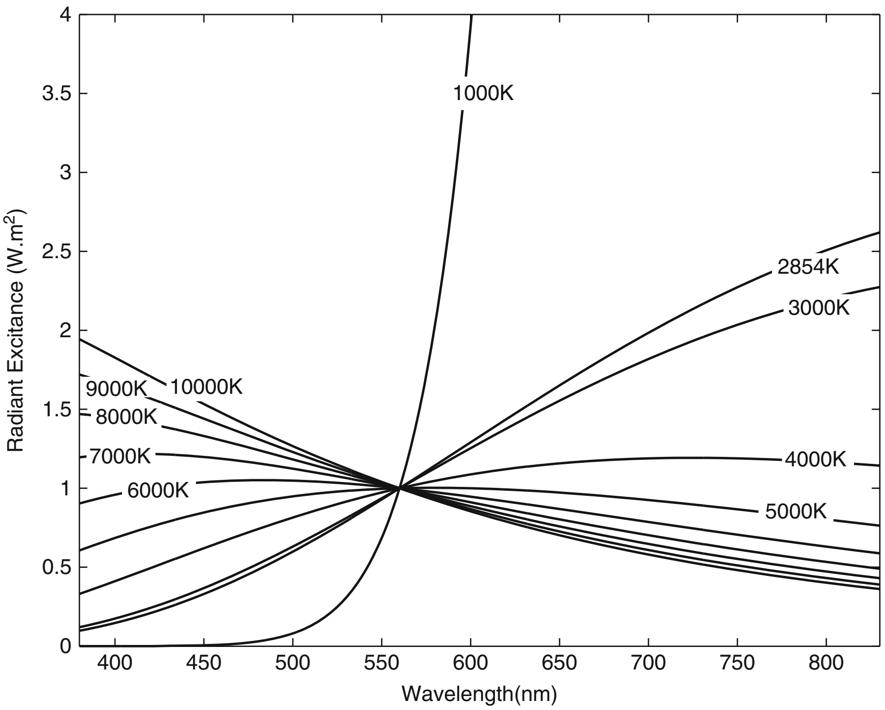
►Planckian Locus

Definition

A blackbody is an idealized object that absorbs all electromagnetic radiation incident on it. The absorbed energy is incandescently emitted with a spectral power distribution that is a function of its temperature. It is called a blackbody partly because it appears black to the human observer when it is cold, as it emits mostly infrared radiation.

Background

Blackbody radiation, in general, stood as a major challenge to the scientists in the nineteenth century as they were pushing the limits of classical physics. Several physicists studied blackbody radiators, including Lord Rayleigh, James Jeans, Josef Stefan, Gustav Kirchhoff, Ludwig Boltzmann, Wilhelm Wien, and finally, Max Planck, who arguably broke the way for quantum physics.



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

Wien's approximation is used to approximately describe the spectral content of radiation from a blackbody. This approximation was first derived by Wilhelm Wien in 1893 [8]. This law was found to be accurate only for short wavelengths of emission spectra of blackbody radiators and is used today only as an approximation. Wien stated that the radiant excitance of a blackbody may be given by

$$M_e(\lambda, T) = \frac{2\pi h\nu^3}{c^2}\exp-\frac{h\nu}{kT} \tag{1}$$

$$= \frac{2\pi hc^2}{\lambda^5}\exp-\frac{hc}{kT\lambda} \tag{2}$$

The above equation holds in the specific case when $\exp-\frac{hc}{kT\lambda} \gg 1$, which typically occurs when the wavelength is short. The work by Wien was soon replaced by the findings of Max Planck, in 1901.

Another important finding regarding blackbody radiators was made by Josef Stefan in 1879 and later formalized by Ludwig Boltzmann. Known as the *Stefan-Boltzmann law*, it states that the total radiant excitance of a blackbody radiator is proportional to the fourth power of its absolute temperature. In other words,

$$\int_{\lambda} M_e(\lambda, T) = \sigma T^4, \tag{3}$$

where $\sigma = 5.674 \times 10^{-8} W \cdot \mathrm{m}^{-1} \cdot \mathrm{K}^{-4}$ is known as the Stefan-Boltzmann constant.

Max Planck, in 1901, stated a more general law, known as *Planck's radiation law*. This describes the spectral distribution of radiant excitance M_e as a function of wavelength (These equations are often given in terms of frequencies instead of wavelength, as shown in this article. This is easily converted back and forth using $\nu \cdot \lambda = c$, where ν denotes frequency, λ the wavelength, and c the speed of light in vacuum.) λ and temperature T and is given by

$$M_e(\lambda, T) = \frac{c_1}{\lambda^5\left[\exp\left(\frac{c_2}{\lambda T}\right) - 1\right]} \tag{4}$$

where $c_1 = 2\pi hc^2 = 3.74183 \times 10^{-16} W \cdot \mathrm{m}^{-2}$, $c_2 = h \cdot c/k = 1.4388 \times 10^{-2}\mathrm{m} \cdot \mathrm{K}$ (c is the speed of light in vacuum: $2.99792458\times10^8\text{m}\cdot\text{s}^{-1}$, h is Planck's constant: $6.62606896\times10^{-34}\text{J}\cdot\text{s}$, k is Boltzmann's constant: $1.38065\times10^{-23}\text{J}\cdot\text{K}^{-1}$) [4], and the excitance is defined in units of $W\cdot\text{m}^{-3}$ [3, 5, 6, 9].

Figure 1 shows the spectral power distribution of various blackbody radiators from 1,000 to 10,000 K, with all spectral power distributions normalized to unity at 560 nm. As the blackbody gets hotter (T increases), one can see that the red content in the spectrum reduces and the blue content increases – an indication of the color as would be seen by the human observer. Both Wien's approximation and the Stefan-Boltzmann law can be derived from Planck's law.

Wien's displacement law may be derived by differentiating the equation for Planck's law Eq. (4) with respect to wavelength λ and equating the result to zero to find the maximum. It states that the spectral distribution of the spectral excitance of a blackbody reaches a maximum at a wavelength λ_m and that the product of the this maximum wavelength and the temperature of a blackbody is a constant, given by:

$$\lambda_m\cdot T=\frac{h\cdot c}{4.965114\,k}=2.8977685\times10^{-3} \tag{5}$$

where h denotes the Planck constant, c denotes the speed of light in vacuum, and k denotes the Boltzmann constant. The corresponding value of the spectral excitance is given by:

$$M_{e\lambda_m}=T^5\;1.286673\times10^{-5}W\text{m}^{-3} \tag{6}$$

Blackbody radiators are the select few sources of illumination that match "standard illuminant" spectral power distributions – this may be seen as valid in the case of the equivalence of standard illuminant "A" and a blackbody with a temperature 2,856 K.

References

1. 1998c CIE (1998) CIE standard illuminants for colorimetry. CIE, Vienna. Also published as ISO 10526/CIE/S006/E1999
2. CIE 15:2004 (2004) Colorimetry. CIE, Vienna
3. Longair MS (1984) Theoretical concepts in physics: an alternative view of theoretical reasoning in physics. Cambridge University Press, Cambridge
4. Mohr PJ, Taylor BN, Newell DB (2007) CODATA recommended values of the fundamental physical constants:2006. National Institute of Standards and Technology, Gaithersburg
5. Planck M (1901) On the law of distribution of energy in the normal spectrum. Annalen der Physik 4:553
6. Rybicki G, Lightman AP (1985) Radiative processes in astrophysics. Wiley-Interscience, New York
7. Wien W (1893) Situngsber. d. Berliner Akad, pp 55–62
8. Wien W (1896) Über die energievertheilung im emissionsspectrum eines schwaarzen körpers. Annalen der Physik 58:662–669
9. Wyszecki G, Stiles WS (1982) Color science: concepts and methods, quantitative data and formulas, 2nd edn. Wiley, New York

Blackbody Radiator

▶Planckian Locus

Blind Deconvolution

Tom Bishop[1] and Paolo Favaro[2]
[1]Image Algorithms Engineer, Apple, Cupertino, CA, USA
[2]Department of Computer Science and Applied Mathematics, Universität Bern, Switzerland

Synonyms

Deblurring; Deconvolution; Kernel estimation; Motion deblurring; PSF estimation

Related Concepts

▶Denoising; ▶Image-Based Modeling; ▶Inpainting

Definition

Blind image deconvolution is the problem of recovering a sharp image (such as that captured by an ideal pinhole camera) from a blurred and noisy one, without exact knowledge of how the image was blurred. The unknown blurring operation may result from camera motion, scene motion, defocus, or other optical aberrations.

Background

A correct photographic exposure requires a trade-off in exposure time and aperture setting. When illumination is poor, the photographer can choose to use a long exposure time or a large aperture. The first setting results in motion blur when the camera moves relative to objects in the scene during the exposure. The second setting results in out-of-focus blur for objects at depths away from the focal plane. Furthermore, these effects may be exacerbated by the user due to camera shake, incorrect focus settings, or other distortions such as atmospheric turbulence.

Under local approximations, these processes can be modeled as convolution operations between an ideal "sharp" image and a *kernel* or point-spread function (PSF). This (PSF) represents how a point of light in the scene would be imaged onto the camera's sensor. In general the (PSF) may be space-varying or depth-varying, such that each point in 3D space has its own response in the kernel function. In general, the effect of the kernel is to blur and remove information from the image (see Fig. 1).

When the (PSF) is known, deconvolution algorithms can be used to remove the effect of these degradations. Deconvolution may be performed using direct (e.g., Fourier-based) or iterative (e.g., gradient descent or conjugate gradient based) algorithms. Essentially, a large linear system must be inverted to recover the sharp image, and depending on the conditioning of the matrix representing the blurring, the solution may be obtained to a greater or lesser accuracy. Observation noise also hinders exact invertibility, and for these reasons *regularization* of the solution is required. Such regularization typically imparts prior knowledge about the expected statistics of the sharp image, such as smoothness, sparseness of its gradients, or compressibility in some domain, and is typically key to obtaining well-behaved solutions.

In practice, the PSF is rarely known from calibration, and in a practical scenario it must be estimated from the blurred image itself. While many algorithms have been proposed to tackle the blind deconvolution problem with some success, a universal solution is not yet available and it is still an active area of research. The difficulty owes in part to the dimensionality of the unknowns in the problem, and its extreme ill-posedness, with the potential for multiple local solutions arising from non-convex optimization problems. Progress has been made in both priors to describe images and blurs and better constrain the solution and estimation methods to better approximate the intractable inference problems.

Related methods also exist to recover the sharp image from multiple observations, for example, a blurred image and a sharp but noisy image, or multiple blurred images, in which case the problem is much more well posed and the solution is more readily obtained; this is also closely related to the problem of super-resolution where the sharp image is estimated at a higher resolution than the input images.

Theory

Imaging Model

The following linear spatially varying (LSV) observation model represents the formation of a general blurred image on the camera sensor:

$$g(\boldsymbol{x}) = \sum_{\substack{\boldsymbol{s}\in\mathcal{S}_f,\\ (\boldsymbol{x},\boldsymbol{s})\in\mathcal{S}_H}} h(\boldsymbol{x},\boldsymbol{s})\, f(\boldsymbol{s}) + w(\boldsymbol{x}), \qquad \boldsymbol{x}\in\mathcal{S}_g, \tag{1}$$

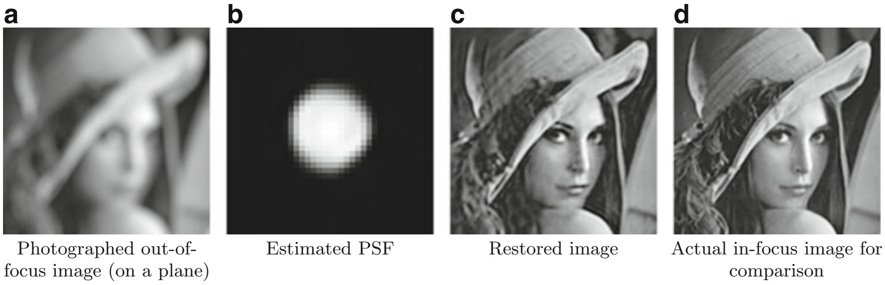
where $f(\boldsymbol{s})$ is the *true* or *sharp* image, $g(\boldsymbol{x})$ the observed blurred image, $w(\boldsymbol{x})$ is additive noise, and $h(\boldsymbol{x},\boldsymbol{s})$ is the kernel function describing the blur. The position $\boldsymbol{x}$ lies in the blurred image support $\mathcal{S}_g \subset \mathbb{R}^2$, while the position $\boldsymbol{s}$ lies in the true image support $\mathcal{S}_f \subset \mathbb{R}^2$. Notice that $\mathcal{S}_H \subset \mathbb{R}^4$ denotes the support of the kernel h. In the case the blur does not vary with position $\boldsymbol{s}$ in the true image and the depth is constant, the kernel may be reduced to a stationary PSF, $h(\boldsymbol{x},\boldsymbol{s}) = h(\boldsymbol{x}-\boldsymbol{s})$. Then the image model becomes a convolution:

$$g(\boldsymbol{x}) = [h \star f](\boldsymbol{x}) + w(\boldsymbol{x}) \tag{2}$$

With discretization and appropriate *lexicographic ordering*, or raster scanning of the images into vectors, either (1) or (2) may also be expressed in matrix-vector form:

$$\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{w}, \qquad \text{or equivalently,} \tag{3}$$

$$\boldsymbol{g} = \boldsymbol{F}\boldsymbol{h} + \boldsymbol{w}. \tag{4}$$


a
b
c
d
Photographed out-of-focus image (on a plane)
Estimated PSF
Restored image
Actual in-focus image for comparison

Blind Deconvolution, Fig. 1 Example of blind deconvolution using the method in [1] where a single image on a plane has been captured by a defocused camera. (**a**) Photographed out-of-focus image (on a plane). (**b**) Estimated PSF. (**c**) Restored image. (**d**) Actual in-focus image for comparison

With the spatially invariant degradation model, the matrices $\boldsymbol{F}$ and $\boldsymbol{H}$ acquire a special structured block form, termed block Toeplitz with Toeplitz blocks (BTTB), with constant block entries on each block diagonal and constant diagonals within each block. Sometimes these matrices are approximated as circulant, implying circular convolution of the sharp image, and then they may be diagonalized using the discrete Fourier transform (DFT), enabling fast calculations to be performed.

Probabilistic Formulation

With uncertainty in the observation model (3), it is natural to estimate the most likely solution for the sharp image using a probabilistic approach. The Bayesian framework provides a unifying way to tackle such ill-posed inverse problems. Here a *likelihood* $\mathrm{p}\,(\boldsymbol{g} \mid \cdot)$ is specified from the imaging model and combined with a *prior* $\mathrm{p}\,(\boldsymbol{f}, \boldsymbol{h} \mid \cdot)$ on the image and blur to be estimated, ensuring that only reasonable solutions are obtained. The resulting posterior distribution

$$\mathrm{p}\,(\boldsymbol{f}, \boldsymbol{h} \mid \boldsymbol{g}, \boldsymbol{\Omega}\,) = \frac{\mathrm{p}\,(\boldsymbol{g} \mid \boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\Omega}\,)\,\mathrm{p}\,(\boldsymbol{f} \mid \boldsymbol{\Omega}\,)\,\mathrm{p}\,(\boldsymbol{h} \mid \boldsymbol{\Omega}\,)}{\mathrm{p}\,(\boldsymbol{g})} \tag{5}$$

is used for inference of the unknowns, where $\boldsymbol{\Omega}$ denotes *hyperparameters* of the model, such as noise variances or regularization parameters; these are usually considered known, but correctly estimating them is often critical for accurate blind deconvolution. The additive noise $\boldsymbol{w}$ is commonly assumed to be Gaussian or sometimes Poisson distributed. In the independent white Gaussian noise (WGN) case with variance σ_w, the distribution is $\mathrm{p}_W\,(\boldsymbol{w}) = \mathcal{N}\,(\boldsymbol{w} \mid 0, \sigma_w \boldsymbol{I}\,)$. Thus, the likelihood of $\boldsymbol{g}$ conditioned on $\boldsymbol{h}, \boldsymbol{f}$ is given by

$$\begin{aligned}\mathrm{p}_G\,(\boldsymbol{g} \mid \boldsymbol{f}, \boldsymbol{h}, \sigma_w\,) &= \mathrm{p}_W\,(\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\,)\\ &= (2\pi\sigma_w)^{-\frac{L_g}{2}} \exp\left[-\frac{1}{2\sigma_w^2}\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\,\|^2\right],\end{aligned} \tag{6}$$

where L_g is the size of the vector $\boldsymbol{g}$.

Bayesian Inference Methods

Due to the complexity of the chosen prior models, it may not be possible to obtain an exact analytic solution of Eq. (5). A common approximation is to compute a point estimate of the unknowns $\boldsymbol{f}$ and $\boldsymbol{h}$ via an optimization procedure. However, these estimates work well only with highly peaked distributions. With more uncertainty in the parameters, it is better to estimate the whole parameters *distribution*. Unfortunately, strategies that do so in the Bayesian framework are typically more computationally demanding [2–4]. Finally, as the estimated posterior distribution of the parameters must have a finite representation, further approximations or simulations must be introduced.

Maximum A Posteriori and Maximum Likelihood

The maximum a posteriori (MAP) solution is one common point estimate where it is possible to prescribe our prior knowledge about the unknowns. It is defined as the values $\hat{\boldsymbol{f}}$, $\hat{\boldsymbol{h}}$, and $\hat{\boldsymbol{\Omega}}$ that maximize the posterior probability density:

$$\begin{aligned}&\{\hat{\boldsymbol{f}}, \hat{\boldsymbol{h}}, \hat{\boldsymbol{\Omega}}\}_{\mathrm{MAP}}\\ &\quad= \operatorname*{argmax}_{\boldsymbol{f},\boldsymbol{h},\boldsymbol{\Omega}} \mathrm{p}\,(\boldsymbol{g} \mid \boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\Omega}\,)\,\mathrm{p}\,(\boldsymbol{f} \mid \boldsymbol{\Omega}\,)\,\mathrm{p}\,(\boldsymbol{h} \mid \boldsymbol{\Omega}\,)\,\mathrm{p}\,(\boldsymbol{\Omega})\,.\end{aligned} \tag{7}$$

A very related method is *maximum likelihood (ML)*, where one looks for

$$\{\hat{\boldsymbol{f}}, \hat{\boldsymbol{h}}, \hat{\boldsymbol{\Omega}}\}_{\mathrm{ML}} = \underset{\boldsymbol{f},\boldsymbol{h},\boldsymbol{\Omega}}{\operatorname{argmax}} \ \mathrm{p}\left(\boldsymbol{g} \mid \boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\Omega}\right). \qquad (8)$$

Notice that the *maximum likelihood (ML)* method is essentially the *maximum a posteriori (MAP)* method where the prior distributions are uniform (uninformative). Despite this equivalence, the *maximum likelihood (ML)* is usually referred to as a non-Bayesian method. The advantage of using the *maximum a posteriori (MAP)* approach is that we can also encode the case where parameters are entirely or partly known by using degenerate distributions (Dirac deltas), that is, $\mathrm{p}(\boldsymbol{\Omega}) = \delta(\boldsymbol{\Omega} - \boldsymbol{\Omega}_0)$. Then, the *MAP* and *ML* formulations become, respectively,

$$\{\hat{\boldsymbol{f}}, \hat{\boldsymbol{h}}\}_{\mathrm{MAP}} = \underset{\boldsymbol{f},\boldsymbol{h}}{\operatorname{argmax}} \ \mathrm{p}\left(\boldsymbol{g} \mid \boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\Omega}_0\right) \mathrm{p}\left(\boldsymbol{f} \mid \boldsymbol{\Omega}_0\right) \mathrm{p}\left(\boldsymbol{h} \mid \boldsymbol{\Omega}_0\right) \qquad (9)$$

$$\{\hat{\boldsymbol{f}}, \hat{\boldsymbol{h}}\}_{\mathrm{ML}} = \underset{\boldsymbol{f},\boldsymbol{h}}{\operatorname{argmax}} \ \mathrm{p}\left(\boldsymbol{g} \mid \boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\Omega}_0\right). \qquad (10)$$

The Bayesian framework can be used to describe several deconvolution methods and to emphasize their differences in terms of choice of likelihood, priors on the image, blur, and hyperparameters. Further differences can be found in how the maximization problem is solved.

A typical example of methods that can be formulated in the Bayesian framework is regularized approaches based on the L_2 norm. One such method is Tikhonov regularization, where a linear system is solved in least-squares sense by introducing an additional L_2 constraint on the unknowns. More in general, the blind deconvolution task is formulated as a constrained minimization where several regularization constraint terms are added.

A common choice is to always use a term in the form of $\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2$, called *data fidelity term*. The additional regularization terms encode the constraints on the unknowns. For example, one may want to impose smoothness of the image and the blur. To do so, a term that penalizes small variations of the image and the blur can be used. As the solution will be a trade-off between the data fidelity term and the regularization terms, regularization parameters are used to adjust their weight.

An important example that illustrates this procedure is [5]. In that work the classical regularized image deconvolution formulation [6, 7] was extended to the blind image deconvolution (BID) case by adding regularization on the blur parameters. The problem is formulated as

$$\hat{\boldsymbol{f}}, \hat{\boldsymbol{h}} = \underset{\boldsymbol{f},\boldsymbol{h}}{\operatorname{argmin}} \Big[\|\boldsymbol{g} - \boldsymbol{H}\boldsymbol{f}\|^2_{\boldsymbol{Q}_w^{-1}} + \lambda_1 \|\boldsymbol{L}_f \boldsymbol{f}\|^2 + \lambda_2 \|\boldsymbol{L}_h \boldsymbol{h}\|^2 \Big], \qquad (11)$$

where λ_1 and λ_2 are the Lagrange multipliers for each constraint, and $\boldsymbol{L}_f$ and $\boldsymbol{L}_h$ are the regularization operators. To avoid oversmoothing the edges each $\boldsymbol{L}$ operator is the Laplacian multiplied by space-varying weights. These weights are obtained from the local image variance as in [7–10].

Alternating Minimization or Iterated Conditional Modes

One of the main difficulties in MAP is to simultaneously recovering both $\boldsymbol{f}$ and $\boldsymbol{h}$. The problem can be already observed in the image model Eq. (1), which is bilinear in both $\boldsymbol{f}$ and $\boldsymbol{h}$. One common way to address this challenge is alternating minimization (AM). When applied to Eq. (11), it performs the minimization by working on one variable at a time while the others are fixed. As a result, the minimization of Eq. (11) becomes a sequence of linear problems. A related method is iterated conditional modes (ICM) proposed by Besag [11].

Minimum Mean-Squared Error

As mentioned earlier on, the MAP estimate is only a point estimate of the whole posterior probability density function (PDF). While this is not a problem when the posterior is highly peaked about the maximum, in the case of high observation noise or a broad (heavy-tailed) posterior, this estimate is likely to be unreliable (as chances of obtaining values that are different from the maximum are very likely). Indeed, in a high-dimensional Gaussian distribution, most of the probability *mass* lies away from the probability *density* peak [12].

One way to correct for this shortcoming is to use the minimum mean-squared error (MMSE) estimate.

The rationale is to find the optimal parameter values as those that minimize the expected mean-squared error between the estimates and the true values. This requires to compute the mean value of $\mathrm{p}(\boldsymbol{f}, \boldsymbol{h}, \boldsymbol{\Omega}|\boldsymbol{g})$. However, computing MMSE estimates analytically is generally difficult. A more practical solution is to use sampling-based methods (see next paragraph).

Markov Chain Monte Carlo Sampling

A general technique to perform inference is to simulate the posterior distribution in Eq. (5), by drawing samples. Provided that we have obtained enough independent samples, this strategy allows us to deal with arbitrarily complex models in high-dimensional spaces, where no analytic solution is available. Markov chain Monte Carlo (MCMC) methods approximate the posterior distribution by the statistics of samples generated from a Markov chain. Widely used Markov chain Monte Carlo (MCMC) algorithms are the Metropolis-Hastings or Gibbs Samplers (see, e.g., [3, 13–15]).

The samples can then be used in Monte Carlo integration to obtain point estimates or other distribution statistics. For instance, in the BID problem the MMSE estimate of the $\boldsymbol{f}$ can be readily obtained by taking the mean of the samples, $\frac{1}{n}\sum_{t=1}^{n}\boldsymbol{f}^{(t)}$.

Markov chain Monte Carlo (MCMC) can provide better solutions than AM or any other method. However, there are some limitations. First, they are very computationally intensive in comparison to the point estimate methods. Second, convergence to the posterior can be theoretically guaranteed, but in practice it can be hard to tell when this has occurred, and it may require a long time to explore the parameter space.

Marginalizing Hidden Variables

In the discussion so far the aim was to recover all the unknowns. However, in most cases one is interested in recovering only the sharp image $\boldsymbol{f}$. This leads to another approach to the BID problem where undesired unknowns are marginalized and inference is performed on the remaining variables (i.e., a subset of $\boldsymbol{f}, \boldsymbol{h}$, and $\boldsymbol{\Omega}$). Therefore, one can approach the BID inference problem in two steps. First, one can calculate

$$\hat{\boldsymbol{h}}, \hat{\boldsymbol{\Omega}} = \underset{\boldsymbol{h},\boldsymbol{\Omega}}{\operatorname{argmax}} \int_{\boldsymbol{f}} \mathrm{p}(\boldsymbol{g}|\boldsymbol{\Omega}, \boldsymbol{f}, \boldsymbol{h})\mathrm{p}(\boldsymbol{f}, \boldsymbol{h}|\boldsymbol{\Omega})\mathrm{p}(\boldsymbol{\Omega})\mathrm{d}\boldsymbol{f} \tag{12}$$

and, second, one selects the sharp image

$$\hat{\boldsymbol{f}}\Big|_{\hat{\boldsymbol{h}},\hat{\boldsymbol{\Omega}}} = \underset{\boldsymbol{f}}{\operatorname{argmax}}\, \mathrm{p}(\boldsymbol{g}|\hat{\boldsymbol{\Omega}}, \boldsymbol{f}, \hat{\boldsymbol{h}})\mathrm{p}(\boldsymbol{f}|\hat{\boldsymbol{\Omega}}). \tag{13}$$

Alternatively, one can also marginalize $\boldsymbol{h}$ and $\boldsymbol{\Omega}$ to obtain

$$\hat{\boldsymbol{f}} = \underset{\boldsymbol{f}}{\operatorname{argmax}} \int_{\boldsymbol{h},\boldsymbol{\Omega}} \mathrm{p}(\boldsymbol{g}|\boldsymbol{\Omega}, \boldsymbol{f}, \boldsymbol{h})\mathrm{p}(\boldsymbol{f}, \boldsymbol{h}|\boldsymbol{\Omega})\mathrm{p}(\boldsymbol{\Omega})\mathrm{d}\boldsymbol{h} \cdot \mathrm{d}\boldsymbol{\Omega}. \tag{14}$$

Deconvolution Under a Gaussian Prior (MAP)

If the point-spread function (PSF) is known and assuming a Gaussian prior $\mathrm{p}\left(\boldsymbol{f} \,\middle|\, \boldsymbol{\Sigma}_f\right) = \mathcal{N}\left(\boldsymbol{f} \,\middle|\, 0, \boldsymbol{\Sigma}_f\right)$ for $\boldsymbol{f}$, with a given covariance matrix $\boldsymbol{\Sigma}_f$, the posterior for $\boldsymbol{f}$ is found as

$$\mathrm{p}\left(\boldsymbol{f} \,|\, \boldsymbol{g}\right) \propto \mathrm{p}\left(\boldsymbol{g} \,|\, \boldsymbol{f}\right)\mathrm{p}\left(\boldsymbol{f} \,\middle|\, \boldsymbol{\Sigma}_f\right) \tag{15}$$

$$\propto \exp\Big(-\frac{1}{2}\big[\boldsymbol{f}^T(\sigma_w^{-2}\boldsymbol{H}^T\boldsymbol{H} + \boldsymbol{\Sigma}_f^{-1}) \boldsymbol{f} - 2\boldsymbol{f}^T(\sigma_w^{-2}\boldsymbol{H}^T\boldsymbol{g}) + \sigma_w^{-2}\boldsymbol{g}^T\boldsymbol{g}\big]\Big) \tag{16}$$

which is a Gaussian

$$\mathrm{p}\left(\boldsymbol{f} \,|\, \boldsymbol{g}\right) \propto \mathcal{N}\left(\boldsymbol{f} \,\middle|\, \boldsymbol{\mu}_{\hat{f}}, \boldsymbol{\Sigma}_{\hat{f}}\right) \tag{17}$$

$$\propto \exp\Big(-\frac{1}{2}\big[\boldsymbol{f}^T\boldsymbol{\Sigma}_{\hat{f}}^{-1}\boldsymbol{f} - 2\boldsymbol{f}^T\boldsymbol{\Sigma}_{\hat{f}}^{-1}\boldsymbol{\mu}_{\hat{f}} + \boldsymbol{\mu}_{\hat{f}}^T\boldsymbol{\Sigma}_{\hat{f}}^{-1}\boldsymbol{\mu}_{\hat{f}}\big]\Big). \tag{18}$$

By comparison of (16) and (18), the parameters are given as

$$\boldsymbol{\Sigma}_{\hat{f}}^{-1} = \sigma_w^{-2}\boldsymbol{H}^T\boldsymbol{H} + \boldsymbol{\Sigma}_f^{-1} \qquad \boldsymbol{\mu}_{\hat{f}} = \boldsymbol{\Sigma}_{\hat{f}}\left(\sigma_w^{-2}\boldsymbol{H}^T\boldsymbol{g}\right). \tag{19}$$

The mean of this distribution, which is also the maximum, is just

$$\hat{\boldsymbol{f}} = \boldsymbol{\mu}_{\hat{f}} = \left(\boldsymbol{H}^T\boldsymbol{H} + \sigma_w^2\boldsymbol{\Sigma}_f^{-1}\right)^{-1}\boldsymbol{H}^T\boldsymbol{g}. \tag{20}$$

In practice, as solving the above equation involves inverting a large linear system, one employs iterative methods.

Application

Blind deconvolution methods are commonly used to restore images that have been distorted by motion blur, out-of-focus blur, and turbulence. As these methods provide an estimate of blur, other uses include digital refocusing, that is, digitally changing the focus setting of a camera after the snapshot, changing the camera bokeh, and obtaining a 3D model of the scene (from the out-of-focus blur).

Acronyms

PSF	point-spread function
MC	Markov chain
MCMC	Markov chain Monte Carlo
ICM	iterated conditional modes
PDF	probability density function
ML	maximum likelihood
MAP	maximum a posteriori
AM	alternating minimization
BID	blind image deconvolution
MMSE	minimum mean-squared error

References

1. Bishop TE, Molina R, Hopgood JR (2008) Blind restoration of blurred photographs via AR modelling and MCMC. In: IEEE international conference on image processing (ICIP), San Diego
2. Gelman A, Carlin JB, Stern HS, Rubin DB (2004) Bayesian data analysis, 2nd edn. Chapman & Hall, London
3. Neal RM (1993) Probabilistic inference using Markov chain Monte Carlo methods. Technical report CRG-TR-93-1, Department of Computer Science, University of Toronto, University of Toronto available online at http://www.cs.toronto.edu/~radford/res-mcmc.html
4. Jordan MI, Ghahramani Z, Jaakola TS, Saul LK (1998) An introduction to variational methods for graphical models. Machine Learning, Kluwer Academic Publishers Hingham, MA, USA, Bari, Italy, 37(2):183–233
5. You YL, Kaveh M (1996) A regularization approach to joint blur identification and image restoration. IEEE Trans Image Process 5(3):416–428
6. Lagendijk RL, Biemond J, Boekee DE (1988) Regularized iterative image restoration with ringing reduction. IEEE Trans Acoust Speech Signal Process 36(12):1874–1887
7. Katsaggelos AK (1985) Iterative image restoration algorithms. PhD thesis, Georgia Institute of Technology, School of Electrical Engineering, Bombai, India
8. Katsaggelos AK, Biemond J, Schafer RW, Mersereau RM (1991) A regularized iterative image restoration algorithm. IEEE Trans Signal Process, Louisville, Kentucky, USA, 39(4):914–929
9. Efstratiadis SN, Katsaggelos AK (1999) Adaptive iterative image restoration with reduced computational load. Machine Learning - The Eleventh Annual Conference on computational Learning Theory archive, Kluwer Academic Publishers Hingham, MA, USA, 37(3):297–336
10. Kang MG, Katsaggelos AK (1995) General choice of the regularization functional in regularized image restoration. IEEE Trans Image Process 4(5):594–602
11. Besag J (1986) On the statistical analysis of dirty pictures. J R Stat Soc B 48(3):259–302
12. Molina R, Katsaggelos AK, Mateos J (1999) Bayesian and regularization methods for hyperparameter estimation in image restoration. IEEE Trans Image Process 8(2): 231–246
13. Andrieu C, de Freitras N, Doucet A, Jordan M (2003) An introduction to MCMC for machine learning. Mach Learn 50:5–43
14. Gilks W, Richardson S, Spiegelhalter D (eds) (1995) Markov chain Monte Carlo in practice: interdisciplinary statistics. Machine Learning, Kluwer Academic Publishers Hingham, MA, USA, 37(2):183–233
15. Ó Ruanaidh JJ, Fitzgerald W (1996) Numerical Bayesian methods applied to signal processing, 1st edn. Springer series in statistics and computing. Springer, New York. ISBN:0-387-94629-2

Blur Estimation

Yu-Wing Tai
Department of Computer Science, Korean Advanced
Institute of Science and Technology (KAIST),
Yuseong-gu, Daejeon, South Korea

Synonyms

Blur Kernel estimation; Point spread function estimation

Related Concepts

▸Defocus Blur; ▸Image Enhancement and Restoration; ▸Motion Blur

Definition

Blur estimation is a process to estimate the point spread function (a.k.a. blur kernel) from an image which suffered from either the motion blur or the defocus blur effects.

Background

When taking a photo with long exposure time, or with wrong focal length, the captured image will look blurry. This is because during the exposure period, the lights captured for a pixel are mixed with the lights captured for the other pixels within a local neighborhood. Such effect is modeled by the point spread function which describes how the lights are mixed during the exposure period.

In motion blur, the point spread function describes the relative motions between the camera and the scene. In defocus blur, the point spread function is related to the distance of a scene point from the focal plane of the camera. Recovering the point spread function is an important step in image deblurring in which the goal is to recover the sharp and clear image from the input blurry image. Also, the estimated point spread function can be used as a feature to evaluate the quality of a photo or to segment the in-focus regions from an image with large depth of field effects.

Theory

Representation

Mathematically, the effects of blurriness can be described by the following equation:

$$B(x,y) = \sum_{(m,n)\in\mathcal{N}(x,y)} I(x-m, y-n)k_{(x,y)}(m,n) \tag{1}$$

where B is the blurry image, I is the sharp image, $k_{(x,y)}$ is the point spread function at (x,y), and $\mathcal{N}$ is the local neighborhood of a pixel, respectively. In general, the point spread function $k_{(x,y)}$ is spatially varying. This means that every pixel can carry different point spread function. This happens when the scene has large depth disparity (for defocus blur) or when the scene contains a moving object (for motion blur) or when the camera exhibits rotational motion (for motion blur) during exposure period.

While the point spread function is spatially varying, the variation of the point spread function is spatially smooth. To simplify the problem, in case of the camera motion, many previous works have assumed the point spread function is spatially invariant, which reduces Eq. (1) into a convolution equation:

$$B = I \otimes k \tag{2}$$

where $\otimes$ is the convolution operator. Recently, [1, 2] generalized Eq. (2) and proposed the projective motion blur model which uses a sequence of homographies to model the camera motion:

$$B(x,y) = \frac{1}{N}\sum_{i=1}^{N} I(H_i[x \;\; y \;\; 1]^t) \tag{3}$$

where H_i is the homography, $[x\ y\ 1]^t$ is the set of homogeneous coordinates of a pixel at (x,y), and N is the number of homographies used to approximate the camera motion in discrete domain. Note that when there is image noise, an additional noise variable will be appended in Eqs. (1)–(3) under the assumption that noise is additive and is independent to the blur process.

Methodology

In order to estimate the point spread function, there are two main methods. The first method relies on hardware modification or calibration. The second method is purely based on software which typically requires assumptions on image prior or regularization to achieve reliable estimation.

If the effect of blur is caused by the camera internal setting, such as lens aberration, the point spread function can be calibrated. To calibrate the point spread function, the simplest method is to capture an image of a spotlight in a dark room. When there is no blur, the image of the captured spotlight (ideally) should occupy only one pixel. When there is blur, the image of the captured spotlight will occupy more than one pixel and the shape of the recorded spotlight is the point spread function. Similarly, for defocus blur, the focal length of the camera can be adjusted in order to obtain a set of defocus point spread function [3]. In the case of the motion blur, Ben-Ezra and Nayer [4] and Tai et al. [5] proposed a hybrid camera system which estimates the point spread function through integration of optical flows from the auxiliary high-speed camera. Yuan et al. [6] use a noisy/blurry image pair to estimate the point spread function for deblurring. Joshi et al. [7] use motion inertia sensor to measure the camera motion in 3D world.

There are many previous works targeting blur estimation using software approaches. In defocus blur, if the point spread function is a Gaussian kernel and the edges are step edges, a defocus map can be obtained through the analysis of edge orientation and edge sharpness [8–10]. The defocus map indicates the scale of the defocus point spread function. In motion deblurring, blur estimation is usually coupled with the motion deblurring process. Fergus et al. [11] and Whyte et al. [2] proposed a multiscale variational Bayesian framework to estimate the point spread function. Jia [12] and Dai et al. [13] analyzed the edge alpha matte to obtain the marginal probability of the point spread function. Shan et al. [14] and Cho and Lee [15] used alternating optimization to iteratively refine the estimated point spread function and the deblurred image. Xu and Jia [16] proposed an edge selection method which improves the performance of blur estimation algorithms. A study on the blind motion deblurring algorithm with analysis on the point spread function can be found in [17].

Application

Blur estimation continuously receives a lot of attention in research area due to its application on deblurring. Since motion blur and defocus blur are common artifacts in imaging system, its applications range from astronomy telescope to satellite imaging, to medical imaging, and to common consumer-level camera. Besides deblurring, point spread function can also be used to evaluate image quality and to identify moving objects from a scene.

References

1. Tai YW, Tan P, Brown M (2011) Richardson-lucy deblurring for scenes under a projective motion path. IEEE Trans PAMI 33(8):1603–1618
2. Whyte O, Sivic J, Zisserman A, Ponce J (2010) Non-uniform deblurring for shaken images. In: IEEE conference on computer vision pattern recognition (CVPR), San Francisco
3. Levin A, Fergus R, Durand F, Freeman WT (2007) Image and depth from a conventional camera with a coded aperture. ACM Trans Graph 26(3):70
4. Ben-Ezra M, Nayar S (2003) Motion deblurring using hybrid imaging. In: IEEE conference on computer vision pattern recognition (CVPR), Madison, vol I, pp 657–664
5. Tai YW, Du H, Brown M, Lin S (2008) Image/video deblurring using a hybrid camera. In: IEEE conference on computer vision pattern recognition (CVPR), Anchorage
6. Yuan L, Sun J, Quan L, Shum H (2007) Image deblurring with blurred/noisy image pairs. 26(3):1
7. Joshi N, Kang S, Zitnick L, Szeliski R (2010) Image deblurring with inertial measurement sensors. ACM Trans Graph 29(3):30
8. Bae S, Durand F (2007) Defocus magnification. Computer Graphics Forum 26(3):571–579 (Proc. of Eurographics)
9. Sun J, Sun J, Xu Z, Shum HY (2008) Image super-resolution using gradient profile prior. In: IEEE conference on computer vision pattern recognition (CVPR), Anchorage
10. Joshi N, Szeliski R, Kriegman D (2008) Psf estimation using sharp edge prediction. In: IEEE conference on computer vision pattern recognition (CVPR), Anchorage
11. Fergus R, Singh B, Hertzmann A, Roweis ST, Freeman WT (2006) Removing camera shake from a single photograph. ACM Trans Graph 25(3):787–794
12. Jia J (2007) Single image motion deblurring using transparency. In: IEEE conference on computer vision pattern recognition (CVPR), Minneapolis
13. Dai S, Wu Y (2008) Motion from blur. In: IEEE conference on computer vision pattern recognition (CVPR), Anchorage
14. Shan Q, Jia J, Agarwala A (2008) High-quality motion deblurring from a single image. ACM Trans Graph 27(3):73
15. Cho S, Lee S (2009) Fast motion deblurring. ACM SIGGRAPH ASIA 28(5):145
16. Xu L, Jia J (2010) Two-phase kernel estimation for robust motion deblurring. In: European conference on computer vision (ECCV), Heraklion
17. Levin A, Weiss Y, Durand F, Freeman W (2009) Understanding and evaluating blind deconvolution algorithms. In: IEEE conference on computer vision pattern recognition (CVPR), Miami

Blur Kernel Estimation

▸Blur Estimation

Body Configuration Recovery

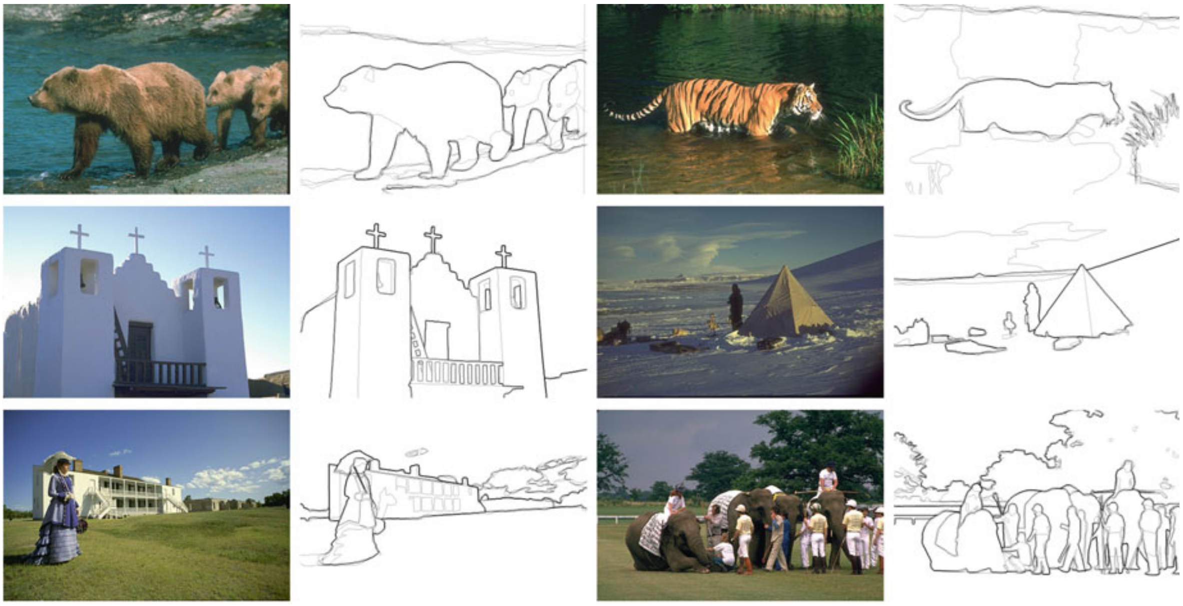
▸Human Pose Estimation

Boundary Detection

Xiaofeng Ren
Intel Science and Technology Center for Pervasive Computing, Intel Labs, Seattle, WA, USA

Synonyms

Boundary extraction; Contour detection



Boundary Detection, Fig. 1 Examples of scenes in the Berkeley Segmentation Dataset [1]. Each photo is labeled by multiple human subjects, and the boundaries are shown as being stacked together. There are variations across human subjects, but the marked boundaries are largely consistent especially for the salient ones

Related Concepts

▸Edge Detection

Definition

Boundary detection is the process of detecting and localizing salient boundaries between objects in a scene.

Background

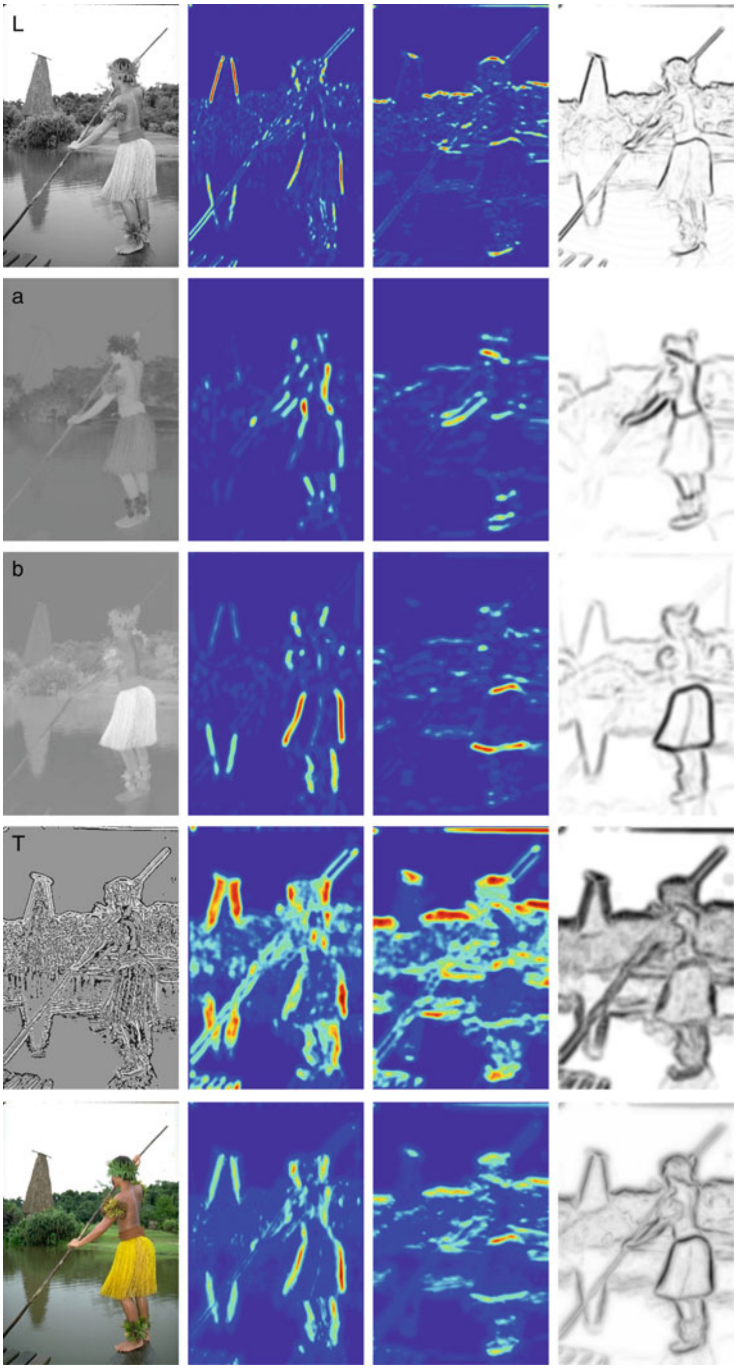
Boundary detection is closely related to, but not identical with, edge detection. Edge detection is a classical problem in computer vision which aims at finding brightness discontinuities. Edge detection is usually viewed as a low-level process of feature extraction that works under the assumption of ideal edge models (such as step and ridge edges).

In comparison, boundary detection is usually viewed as a mid-level process of finding boundaries of (and between) objects in scenes, thus having close ties with both grouping/segmentation and object shape. A large-scale dataset of natural images with human-marked groundtruth boundaries, the *Berkeley Segmentation Dataset* (BSDS) [1, 2], was established in 2001 and quickly became the standard benchmark for both boundary detection and segmentation (see examples in Fig. 1). The Berkeley Segmentation Dataset helped defining the problem of boundary detection and clarifying several fundamental issues:

1. It directly addressed the complexities of real-world scenes by using a variety of photos from the Corel database.
2. It defined boundary detection as a perceptual problem by using human-marked boundaries as the groundtruth.
3. It showed that boundary detection is well defined by demonstrating that boundaries marked by human subjects are consistent.
4. It illustrated many challenges of boundary detection, including those of real-world texture, complex object appearance, and low-contrast boundaries.

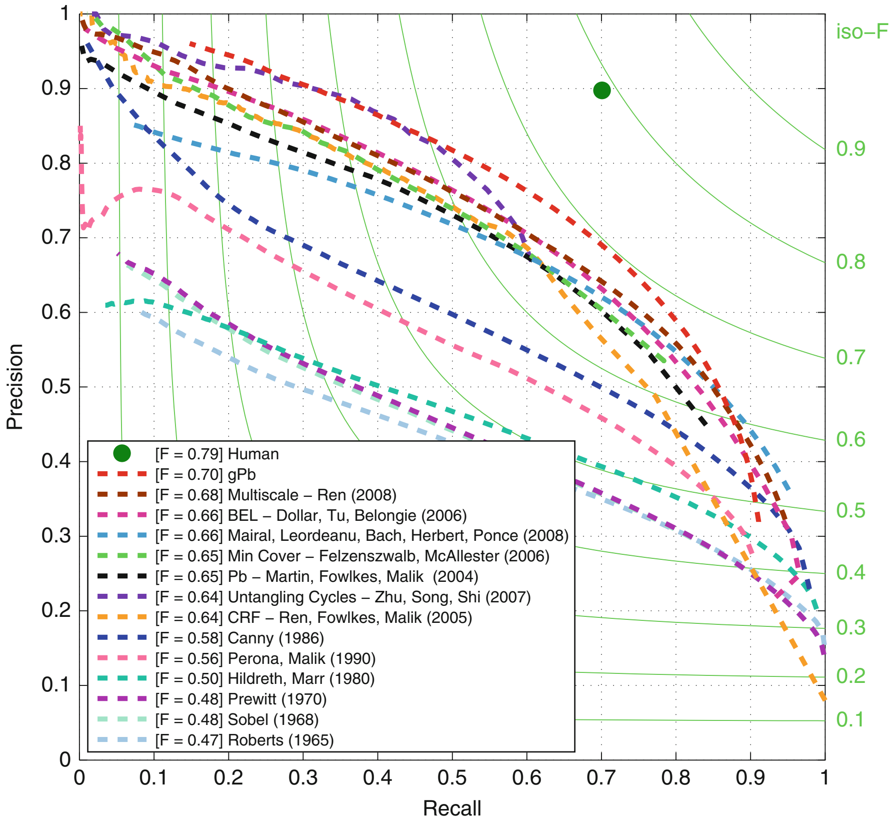
By clarifying the task and establishing quantitative evaluation metrics, the Berkeley benchmark has witnessed and motivated large progresses in boundary detection in the recent years.

Boundary Detection, Fig. 2 Boundary detection combines multiple types of contrast (Courtesy of [9], see details there). Top-performing boundary detectors integrate together local contrast measurements from multiple channels (row 1, brightness; 2 and 3, color; and 4, texture, through textons) and multiple orientations (column 2, vertical; and 3, horizontal). *Red* color means high probability being a boundary and *blue* otherwise. The combined boundary contrast (last row, last column) is much better than any individual channel



Local Boundary Detection

Early approaches to edge detection used local derivative filters such as the Roberts, Sobel, or Prewitt filters [3]. More advanced solutions included that of the zero crossing of Marr and Hildreth [4], the optimal filter design and non-maximum suppression in the Canny detector [5], and the use of quadrature filter pairs in oriented energy [6]. Scale (of the filter) is an important issue in edge detection and



Boundary Detection, Fig. 3 Precision-recall curves and F-measures of classical and modern boundary detection algorithms on the Berkeley benchmark (Courtesy of [9], also see [1]). A variety of approaches have been proposed and evaluated on the benchmark. One can typically observe qualitative improvements in the boundary detection accuracy as the F-measure increases. State-of-the-art boundary detectors perform much better than, for instance, the Canny detector

Lindeberg proposed a mechanism for automatic scale selection [7].

The key concept in boundary detection is that of *contrast*: regions on two sides of a boundary tend to have different appearances; consequently, there tends to be a high contrast at a boundary location. To a large extent, this contrast can be captured and measured locally in an image neighborhood (e.g., a disk with a fixed radius). Local contrast can be measured in a number of ways, such as using linear filters or computing distances between histograms. To handle real-world scenes, modern boundary detectors utilize contrast information from multiple channels (including brightness, color, and texture), multiple orientations, and multiple scales (see examples in Fig. 2). Good examples of these contrast operators can be found in the *Pb* work (probability-of-boundary) of Martin et al. [8] and the *gPb* work (global probability-of-boundary) of Arbelaez et al. [9].

Given the complexities of contrast cues and the availability of labeled images, local boundary detection is often formulated as learning a binary classifier of boundary vs. non-boundary, which will produce a soft boundary "likelihood" at each pixel. Such a dense boundary map can be used directly or converted to a sparse boundary map through non-maximum suppression. A number of supervised machine learning techniques were used and tested in [8] to combine a small set of handcrafted contrast cues. Others have taken a more direct learning approach, such as using boosting trees to combine thousands of simple features over patches [10].

Global Boundary Detection

Boundaries are not local phenomena that occur independently at pixels. In fact, boundaries are defined at the object level, and boundary pixels tend to form long, smooth contours, as evident from the examples in Fig. 1. Considerable efforts have been devoted to extracting boundaries globally, closely related to the classical problem of contour completion in perceptual organization. Algorithms for global boundary detection can be quantitatively evaluated based on their precision recall on the boundary detection task, same as for local boundary detection.

A variety of very different formulations have been proposed for global boundary detection and contour extraction, including classical works such as the Mumford-Shah functional [11]. Several recent approaches have successfully demonstrated, through benchmarking, that globalization greatly improves boundary detection accuracy over local detectors. Ren et al. [12] applied constrained Delaunay triangulation (CDT) to decompose locally detected contours into pieces and used conditional random fields (CRF) and belief propagation to integrate local contrast cues through interactions at junctions. Zhu et al. [13] computed complex eigenvectors of a normalized random walk matrix, using circular embedding, to detect topologically closed cycles. In the *gPb* work of Arbelaez et al. [9], eigenvectors of the affinity matrix were first computed, as in Normalized Cuts [14], and then the gradients of these eigenvectors were added to the local contrast cues to produce a single contrast map.

[Optional]: One related but different form of global boundary detection can be found in the case of *top-down object segmentation* [15], where the algorithm has access to the knowledge (such as shape or texture) of the objects that are in the scene. It is beyond the scope of the discussion here, as boundary detection typically refers to the *bottom-up* case where no high-level object knowledge is needed.

Application

Boundary detection is fundamentally connected to both image segmentation and object shape, and there should be no surprise that advances in boundary detection have led to many interesting applications in segmentation and object recognition.

For image segmentation, the use of *intervening contour* [16] allows one to convert any boundary map to pairwise affinities for use in the Normalized Cuts framework, and many systems (e.g., [17]) have been using modern boundary detectors such as the *Pb* operator [8]. *Pb* is also used in [18], combined with the Watershed algorithm, to produce superpixels. Arbelaez et al. [9] proposed a hierarchical segmentation algorithm that, using the *gPb* boundary operator, produces compelling segmentation results and at the same time further improves the boundary detection accuracy.

For object recognition, boundary detectors such as *Pb* and *gPb* are often used to produce a boundary map, which is in turn used to compute shape descriptors. For instance, the work of Berg et al. [19] used *Pb* boundary maps with Geometric Blur for object and face recognition. The work of Ferrari et al. [20] used *Pb* to produce contour segments as a basis for shape matching. There are many segmentation-based approaches to recognition that also heavily rely on the quality of boundary detection (e.g., [21]).

State-of-the-art boundary detectors are sophisticated and require fairly intensive computation, which limits their applicability. There have been studies and efforts to speed up boundary detectors. In particular, the GPU-based detector of Catanzaro et al. [22] achieved a two-orders-of-magnitude improvement of speed over *gPb* without suffering any loss in boundary quality.

References

1. Martin D, Fowlkes C, Malik J (2002) Berkeley segmentation dataset. http://www.cs.berkeley.edu/projects/vision/bsds
2. Martin D, Fowlkes C, Tal D, Malik J (2001) A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of international conference on computer vision (ICCV), Vancouver, vol 2, pp 416–423
3. Faugeras O (1993) Three-dimensional computer vision: a geometric viewpoint. MIT, Cambridge
4. Marr D, Hildreth E (1980) Theory of edge detection. Proc R Soc Lond B 207(1167):187–217
5. Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8:679–698
6. Morrone M, Owens R (1987) Feature detection from local energy. Pattern Recognit Lett 6:303–13
7. Lindeberg T (1998) Edge detection and ridge detection with automatic scale selection. Int J Comput Vis 30: 117–156
8. Martin D, Fowlkes C, Malik, J (2004) Learning to detect natural image boundaries using local brightness, color and texture cues. IEEE Trans Pattern Anal Mach Intell 26(5): 530–549

9. Arbelaez P, Maire M, Fowlkes C, Malik J (2010) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33(5):898–916
10. Dollar P, Tu Z, Belongie S (2006) Supervised learning of edges and object boundaries. In: Proceedings of computer vision and pattern recognition (CVPR), New York, vol 2, pp 1964–1971
11. Mumford D, Shah J (1989) Optimal approximation by piecewise smooth functions and associated variational problems. Commun Pure Appl Math 42:577685
12. Ren X, Fowlkes C, Malik J (2008) Learning probabilistic models for contour completion in natural images. Int J Comput Vis 77(1–3):47–64
13. Zhu Q, Song G, Shi J (2007) Untangling cycles for contour grouping. In: Proceedings of international conference on computer vision (ICCV), Vancouver, pp 1–8
14. Malik J, Belongie S, Leung T, Shi J (2001) Contour and texture analysis for image segmentation. Int J Comput Vis 43(1):7–27
15. Borenstein E, Ullman S (2002) Class-specific, top-down segmentation. In: Proceedings of european conference on computer vision (ECCV), Copenhagen, vol 2, pp 109–124
16. Leung T, Malik J (1998) Contour continuity in region-based image segmentation. In: Proceedings of european conference on computer vision (ECCV), Freiburg, Germany. vol 1, pp 544–559
17. Mori G, Ren X, Efros A, Malik J (2004) Recovering human body configurations: combining segmentation and recognition. In: Proceedings of computer vision and pattern recognition (CVPR), Washington, vol 2, pp 326–333
18. Hoiem D, Efros A, Hebert M (2007) Recovering occlusion boundaries from a single image. In: Proceedings of international conference on computer vision (ICCV), Rio de Janeiro
19. Berg A, Berg T, Malik J (2005) Shape matching and object recognition using low distortioncorrespondence. In: Proceedings of computer vision and pattern recognition (CVPR), San Diego, vol 1, pp 26–33
20. Ferrari V, Tuytelaars T, Gool LV (2006) Object detection by contour segment networks. In: Proceedings of european conference on computer vision (ECCV), Graz, pp 14–28
21. Gu C, Lim J, Arbeláez P, Malik J (2009) Recognition using regions. In: Proceedings of computer vision and pattern recognition (CVPR), Miami
22. Catanzaro B, Su BY, Sundaram N, Lee Y, Murphy M, Keutzer K (2009) Efficient, high-quality image contour detection. In: Proceedings of international conference on computer vision (ICCV), Kyoto, pp 2381–2388
23. Maire M, Arbelaez P, Fowlkes C, Malik J (2008) Using contours to detect and localize junctions in natural images. Proceedings of computer vision and pattern recognition (CVPR), Anchorage, pp 1–8

Boundary Extraction

►Boundary Detection

BRDF Measurement

►Recovery of Reflectance Properties

BRDF Models

►Reflectance Models