

S

Saturation (Imaging)

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

Synonyms

[Clipping](#)

Related Concepts

► [Radiometric Response Function](#)

Definition

In imaging, saturation is a type of distortion where the recorded image is limited to some maximum value, interfering with the measurement of bright regions of the scene.

Background

The role of a sensor element is to measure incident irradiance and record that quantity as an image intensity value. However, physical constraints limit the maximum irradiance that can be measured for a given camera setting. In the absence of noise, the mapping from irradiance to image intensity is fully described by the *radiometric response function*, a monotonically

increasing function whose range is restricted by the maximum irradiance. Pixels whose intensity corresponds to this maximum are known as saturated.

Saturated pixels contain less information about the scene than other pixels. While non-saturated pixels can be related to the incident irradiance by applying the inverse of the radiometric response function, saturated pixels provide only a lower bound on irradiance. Therefore, estimating the irradiance of saturated pixels is similar to other image “hallucination” tasks such as inpainting [2].

Since many computer vision algorithms assume a linear relationship between sensor irradiance and the measured image intensity, it is important to identify saturated pixels and handle them appropriately. In practice, saturated pixels are often treated as missing values or otherwise ignored.

Theory

In the idealized noise-free case, the image intensity M of a pixel can be described as mapping the incident irradiance I according to the radiometric response function $f(\cdot)$, limited by the maximum irradiance I_{\max} ,

$$M = f(\min(I, I_{\max})). \quad (1)$$

For an irradiance of I_{\max} or higher, the image intensity will saturate at its maximum value of $M_{\max} = f(I_{\max})$. Since saturated pixels do not have unique corresponding irradiance values, they provide no direct information about incident irradiance beyond imposing a lower bound of I_{\max} .

Identifying saturated pixels is straightforward in practice, since many saturated pixels have the nominal maximum pixel value of M_{\max} . Sensor noise and other on-camera processing introduce the minor complication that saturated pixels may have values slightly less than this maximum. This effect is easily addressed, however, by using a lower, more conservative threshold to detect saturation [6, 8].

Saturation is caused by underlying physical characteristics of the sensor which limit the highest irradiance that can be measured for the given settings of the camera. In a digital sensor, where incident photoelectrons are recorded as electric charge, each sensor element can store a maximum amount of charge known as the *full well capacity*. Together with the exposure time and amplifier gain, the full well capacity imposes a limit on the maximum irradiance that can be measured before saturation. Film-based sensors are subject to saturation as well, but the mechanism limiting their photosensitivity is chemical [9].

While modern digital sensors are designed to dissipate excess charge above the full well capacity, for very bright parts of the scene, excess charge from a saturated pixel can spill over to adjacent regions. This artifact, known as *blooming*, can lead to saturation in pixels that would not otherwise be saturated.

Application

Saturation can pose problems for computer vision algorithms that assume linearity unless saturated pixels are identified and handled appropriately. For example, methods operating in the Fourier domain require special attention to saturation [1, 18], because the global nature of the transform means that even isolated saturated pixels can corrupt the whole image. The two main approaches to dealing with saturated pixels are explicitly treating them as missing values and interpolating them from surrounding pixels.

The effect of saturation should also be taken into account when estimating the parameters of sensor noise from an image [5, 11]. Pixels near saturation will demonstrate reduced variance in general, since the maximum value imposed by saturation will make their samples closer on average.

From the standpoint of photography, camera settings should be chosen to avoid saturation in

regions of interest, otherwise important detail or color information may be lost. Photographers describe saturated images as being *overexposed* or as having clipped or blown highlights. Although ill-posed, a problem of great practical interest to photographers is recovering detail in saturated regions of the scene or at least hallucinating plausible detail.

Under mild overexposure, only partial color information may be lost due to saturation. Partial saturation results from the different spectral sensitivities of each color channel, leading one channel to saturate before the others. In this setting, the main approach for restoring detail is to represent the correlation between color channels, using either global [19] or spatially varying [7, 12] color distribution models, then using this correlation to transfer information from the non-saturated color channels.

With greater overexposure, pixels become saturated in all color channels. The most common approach for restoring detail in this setting is to blindly extrapolate smooth peaks within saturated regions [7, 15, 17]. In fact, saturated regions can sometimes provide quantitative evidence about the underlying irradiance. Provided that overexposure is moderate and the scene is sufficiently smooth, the band-limitation of irradiance [1] or the resulting noise distribution [4] can be exploited to recover detail in fully saturated regions. For more severe overexposure or larger saturated regions, none of these methods are generally sufficient. In such cases, user guidance may be enlisted to help transfer plausible high-frequency detail from other sources [17].

In general, choosing the exposure setting for a photo requires balancing competing goals. While overexposure causes loss of detail in the highlights due to saturation, underexposure leads to higher relative noise. The relationship between noise and saturation defines the *dynamic range* of the sensor and determines the range of irradiances that can be captured acceptably in a single shot. When restricted to a single shot, one should generally choose the exposure setting so that the brightest region of interest falls just below the saturation point [14].

For scenes with large dynamic range, such considerations have motivated *high dynamic range imaging* methods based on capturing multiple photos with different exposure times [3], each of which saturates at a different irradiance. There is also an ongoing effort to develop new kinds of high dynamic range sensors

offering higher effective saturation levels [10]. A broad range of new designs have been proposed, including sensors that record the precise length of exposure time needed to reach saturation and sensors with a logarithm-like response. Each of these designs presents unique tradeoffs, including different noise characteristics over their operating range [10]. An orthogonal imaging approach is to use spatial multiplexing to incorporate multiple types of sensor elements, each having different sensitivities [13, 18] or sizes [16].

References

1. Abel JS, Smith JO (1991) Restoring a clipped signal. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing, Toronto, pp 1745–1748
2. Criminisi A, Pérez P, Toyama K (2003) Object removal by exemplar-based inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Madison, vol 2, pp 721–728
3. Debevec PE, Malik J (1997) Recovering high dynamic range radiance maps from photographs. In: Proceedings of the ACM SIGGRAPH, Los Angeles, pp 369–378
4. Foi A (2009) Clipped noisy images: heteroskedastic modeling and practical denoising. *Signal Process* 89(12): 2609–2629
5. Foi A, Trimeche M, Katkovnik V, Egiazarian K (2008) Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Trans Image Process* 17(10):1737–1754
6. Granados M, Adjin B, Wand M, Theobalt C, Seidel H-P, Lensch HPA (2010) Optimal HDR reconstruction with linear digital cameras. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 215–222
7. Guo D, Cheng Y, Zhuo S, Sim T (2010) Correcting over-exposure in photographs. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 515–521
8. Hasinoff SW, Durand F, Freeman WT (2010) Noise-optimal capture for high dynamic range photography. In: Proceedings IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 553–560
9. Hunt RWG (2004) The reproduction of colour, 6th edn. Wiley, Hoboken
10. Kavusi S, El Gamal A (2004) Quantitative study of high dynamic range image sensor architectures. In: Proceedings of the SPIE electronic imaging conference, San Jose, CA, vol 5301, pp 264–275
11. Liu C, Szeliski R, Kang SB, Zitnick CL, Freeman WT (2008) Automatic estimation and removal of noise from a single image. *IEEE Trans Pattern Anal Mach Intell* 30(2):299–314
12. Masood SZ, Zhu J, Tappen MF (2009) Automatic correction of saturated regions in photographs using cross-channel correlation. *Proc Pac Graph* 28:1861–1869
13. Narasimhan SG, Nayar SK (2005) Enhancing resolution along multiple imaging dimensions using assorted pixels. *IEEE Trans Pattern Anal Mach Intell* 27(4):518–530
14. Reichmann M (2003) <http://www.luminous-landscape.com/tutorials/expose-right.shtml>
15. Rempel AG, Trentacoste M, Seetzen H, Young HD, Heidrich W, Whitehead L, Ward G (2007) Ldr2hdr: on-the-fly reverse tone mapping of legacy video and photographs. *ACM Trans Graph* 26
16. Sugimoto M (2008) Digital camera. United States Patent 7,468,746, Dec 2008
17. Wang L, Wei L-Y, Zhou K, Guo B, Shum H-Y (2007) High dynamic range image hallucination. In: Proceedings of the eurographics symposium on rendering, Grenoble, pp 1–7
18. Wetzstein G, Ihrke I, Heidrich W (2010) Sensor saturation in fourier multiplexed imaging. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), San Francisco, pp 545–552
19. Zhang X, Brainard DH (2004) Estimation of saturated pixel values in digital color imaging. *J Opt Soc Am A* 21(12):2301–2310

Scale Selection

Tony Lindeberg

School of Computer Science and Communication,
KTH Royal Institute of Technology, Stockholm,
Sweden

Synonyms

Automatic scale selection; Scale-invariant image features and image descriptors

Related Concepts

► [Corner Detection](#); ► [Edge Detection](#)

Definition

The notion of scale selection refers to methods for estimating characteristic scales in image data and for automatically determining locally appropriate scales in a scale-space representation, so as to adapt subsequent processing to the local image structure and compute scale-invariant image features and image descriptors.

An essential aspect of the approach is that it allows for a bottom-up determination of inherent scales of features and objects without first recognizing them or

delimiting, alternatively segmenting, them from their surroundings.

Scale selection methods have also been developed from other viewpoints of performing noise suppression and exploring top-down information.

Background

The concept of *scale* is essential when computing features and descriptors from image data. Real-world objects may contain different types of structures at different scales and may therefore appear in different ways depending on the scale of observation. When observing objects by a camera or an eye, there is an additional scale problem due to perspective effects, implying that distant objects will appear smaller than nearby objects. A vision system intended to operate autonomously on image data acquired from a complex environment must therefore be able to handle and be robust to such scale variations.

For a vision system that observes an unknown scene, there is usually no way to a priori know what scales are appropriate for extracting the relevant information. Hence, a multi-scale representation of the image data is essential, whereby the original signal is embedded into a *one-parameter family* of signals using scale as the parameter. Given an N -dimensional signal $f: \mathbb{R}^N \rightarrow \mathbb{R}$ and with the notation $x = (x_1, \dots, x_N) \in \mathbb{R}^N$, the *scale-space representation* [8, 10, 26] of f is defined by the convolution operation

$$L(x; t) = \int_{\xi \in \mathbb{R}^N} f(x - \xi) g(\xi; t) d\xi, \quad (1)$$

where $g: \mathbb{R}^N \times \mathbb{R}_+ \rightarrow \mathbb{R}$ denotes the *Gaussian kernel*

$$g(x; t) = \frac{1}{(2\pi t)^{N/2}} e^{-|x|^2/2t} \quad (2)$$

and the variance $t = \sigma^2$ of this kernel is referred to as the *scale parameter*. Based on this representation, *Gaussian derivatives*, or *scale-space derivatives*, at any scale t can then be computed by differentiating the scale-space representation or equivalently by convolving the original image with Gaussian derivative kernels:

$$L_{x^\alpha}(\cdot; t) = \partial_{x^\alpha} L(\cdot; t) = (\partial_{x^\alpha} g(\cdot; t)) * f(\cdot) \quad (3)$$

(with multi-index notation $\alpha = (\alpha_1, \dots, \alpha_N)$ for $\partial_{x^\alpha} = \partial_{x_1}^{\alpha_1} \dots \partial_{x_N}^{\alpha_N}$). Such Gaussian derivatives can be used as a basis for expressing a large number of visual modules including feature detection, feature classification, image matching, motion, shape cues, and image-based recognition [14].

Theory

The notion of scale selection complements traditional *scale-space theory* by providing explicit mechanisms for generating hypotheses about interesting scales.

Scale Selection from γ -Normalized Derivatives

A particularly useful methodology for computing estimates of *characteristic scales* is by detecting *local extrema over scales* of differential expressions in terms of *γ -normalized derivatives* [11, 12] defined by

$$\partial_\xi = t^{\gamma/2} \partial_x. \quad (4)$$

A general and very useful property of this construction is that if two signals f and f' are related by a scaling transformation

$$f'(x') = f(x) \quad \text{with} \quad x' = s x, \quad (5)$$

and if there is a local extremum over scales at $(x_0; t_0)$ in a differential expression $\mathcal{D}_{\gamma\text{-norm}} L$ defined as a homogeneous polynomial of Gaussian derivatives computed from the scale-space representation L of the original signal f , then there will be a corresponding local extremum over scales at $(x'_0; t'_0) = (s x_0; s^2 t_0)$ in the corresponding differential expression $\mathcal{D}_{\gamma\text{-norm}} L'$ computed from the scale-space representation L' of the rescaled signal f' [11, Sect. 4.1].

This scaling result holds for all homogeneous polynomial differential expressions, including rotationally invariant differential invariants, and implies that local extrema over scales of γ -normalized derivatives are preserved under scaling transformations. Thereby, such local extrema over scales provide a theoretically well-founded way to automatically adapt the scale levels to local scale variations.

Specifically, scale-normalized scale-space derivatives of order $|\alpha| = \alpha_1 + \dots + \alpha_N$ at corresponding

points will then be related according to

$$L'_{\xi^{\alpha}}(x'; t') = s^{|\alpha|(\gamma-1)} L_{\xi^{\alpha}}(x; t), \quad (6)$$

which means that $\gamma = 1$ implies perfect scale invariance in the sense that the γ -normalized derivatives at corresponding points will be equal. If $\gamma \neq 1$, the difference in magnitude can on the other hand be easily compensated for using the scale values of corresponding scale-adaptive image features.

These results imply that detection of image features and computation of image descriptors at scale levels equal to or proportional to the scales at which there are local extrema over scales constitutes a very general methodology for obtaining *scale-invariant image features* and *scale-invariant image descriptors*.

Indeed, it can also be axiomatically shown that the notion of γ -normalized derivatives arises by necessity, given the condition that local extrema over scales of scale-normalized derivatives should be preserved under scaling transformations [11, Appendix A.1].

Relation to Frequency Estimation

There is a conceptual similarity between this principle and local frequency estimation from peaks in the Fourier transform. For a one-dimensional sine wave

$$f(x) = \sin(\omega x), \quad (7)$$

it can be shown [11, Sect. 3] that there will be a peak in the magnitude of the m th order γ -normalized derivative at a scale

$$\sigma_{\max} = \frac{\sqrt{\gamma m}}{2\pi} \lambda \quad (8)$$

proportional to the wavelength $\lambda = 2\pi/|\omega|$ of the signal. Two conceptual differences compared to Fourier-based frequency estimation, however, are that (i) no window size is needed for computing the Fourier transform and (ii) this approach applies also to nonlinear differential expressions.

Relations to Image Statistics

It can be shown [11, Sect. 9.1] that the notion of γ -normalized derivatives corresponds to normalizing the m th order N -dimensional Gaussian derivatives to

constant L_p -norms over scale with

$$p = \frac{1}{1 + \frac{m}{N}(1 - \gamma)}, \quad (9)$$

where the perfectly scale-invariant case $\gamma = 1$ corresponds to L_1 -normalization for all orders m .

It can also be shown [11, Sect. 9.2] that the γ -normalized derivatives are *neutral* with respect to *self-similar* power spectra of the form

$$S_f(\omega) = |\omega|^{-N-2m(1-\gamma)}. \quad (10)$$

Natural images often show a qualitative behavior similar to this [5].

Scale-Space Signatures

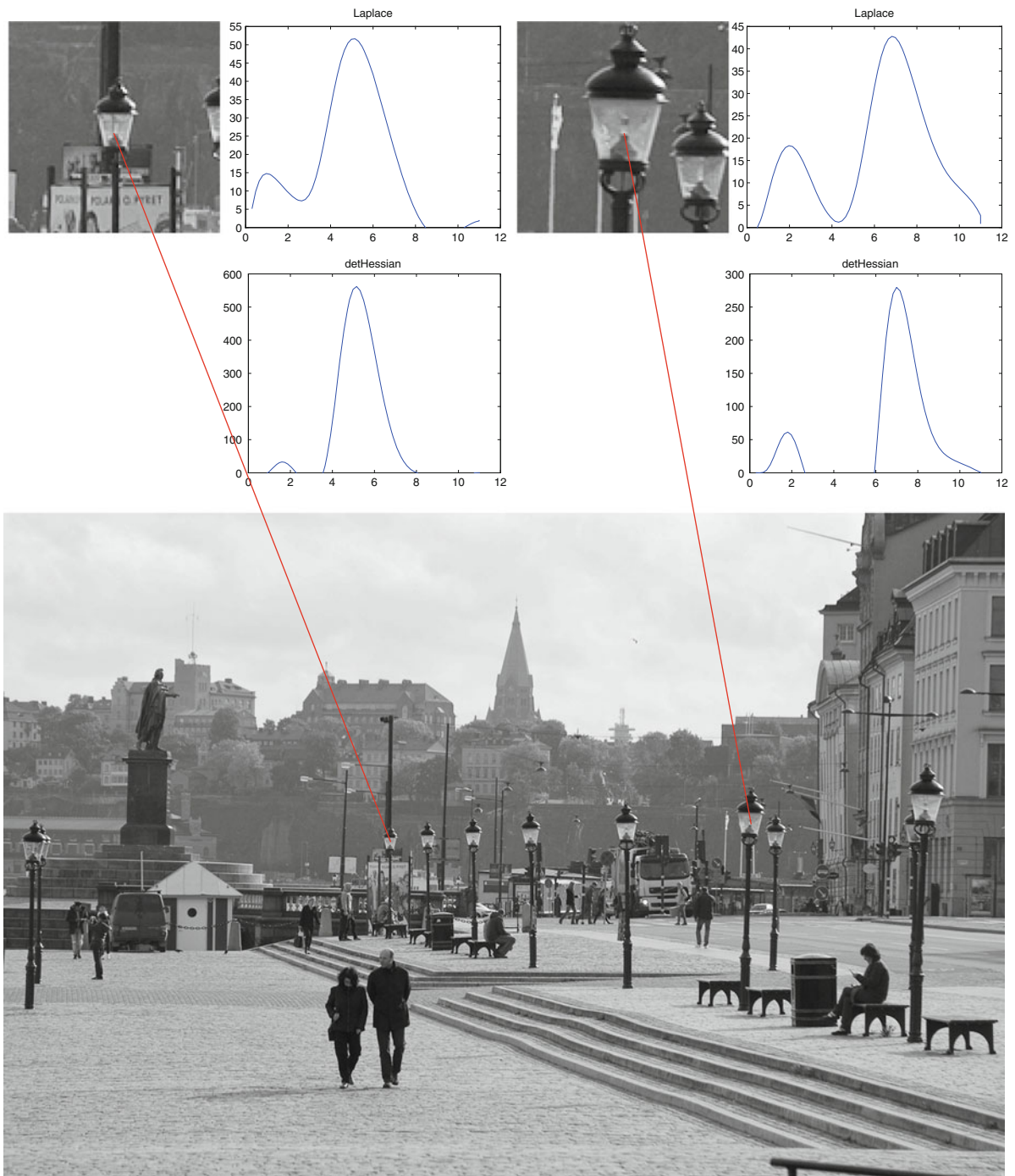
Figure 1 illustrates the basic idea, by showing the so-called *scale-space signatures* accumulated in the two-dimensional case (In the specific 2-D case, the simplifying notation $(x, y) \in \mathbb{R}^2$ is used instead of $x = (x_1, x_2) \in \mathbb{R}_2$, implying that $L_{x_1 x_1} = L_{xx}$, $L_{x_1 x_2} = L_{xy}$, $L_{x_2 x_2} = L_{yy}$, etc.) for two generally applicable differential entities for scale selection: the *scale-normalized Laplacian* [10, Sect. 13.3] [11, Sect. 5] (with $\gamma = 1$)

$$\nabla^2 L_{\text{norm}} = t (L_{xx} + L_{yy}) \quad (11)$$

and the *scale-normalized determinant of the Hessian* [10, 11] (also with $\gamma = 1$)

$$\det \mathcal{H}_{\text{norm}} L = t^2 (L_{xx} L_{yy} - L_{xy}^2). \quad (12)$$

In the scene in Fig. 1, there are strong perspective scaling effects due to differences in depth between similar objects in the world. These scale variations are reflected in the scale-space signatures in the respect that the local extrema over scales are assumed at finer scales for distant objects and at coarser scales for nearby objects. If one computes the ratio between the scale values in terms of a scale parameter $\sigma = \sqrt{t}$ of dimension [length], then the ratio between the scale values is in very good agreement with the ratio between the sizes of the objects in the image domain as measured by a ruler. This property illustrates one of the scale-invariant properties of the scale selection mechanism.



Scale Selection, Fig. 1 *Scale-space signatures* accumulated for image structures of different size in the image domain. The *upper part* of the illustration shows windows around two details from the image at the *bottom*, with corresponding scale-space signatures of the scale-normalized Laplacian $\nabla_{norm}^2 L$ and the scale-normalized determinant of the Hessian $\det \mathcal{H}_{norm} L$ accumulated at the central point in each window. As can be seen from the graphs, the local extrema over scales are assumed at coarser

scales for the larger size object than for the smaller size object. Specifically, the ratio between the scale values at which the local extrema are assumed provides an estimate of the relative amount of scaling, when measured in dimension [length] (In the graphs, the horizontal axis represents *effective scale* [10, pp 180–182] approximated by $\tau \approx \log_2(1 + t)$) (Reprinted from [27] with permission)

General Framework for Defining Scale-Invariant Image Descriptors

By computing an image descriptor at a scale proportional to the detection scale \hat{t} of a scale-invariant image feature or by normalizing an image patch by a corresponding scaling factor $\hat{\sigma} = \sqrt{\hat{t}}$ provides two very general scale normalization mechanisms that can be used for defining much wider classes of *scale-invariant image descriptors* [11, 15] (see the “Applications” section below for two specific examples regarding image-based recognition). The scale-invariant properties of these descriptors originate from the general scale-invariant property of local extrema over scales of differential expressions in terms of γ -normalized derivatives.

Figure 2 illustrates how scale normalization can be performed in this way by rescaling the local image patches around the two details in Fig. 1 using the scale values $\hat{\sigma} = \sqrt{\hat{t}}$ at which the Laplacian $\nabla_{norm}^2 L$ and the determinant of the Hessian, respectively, assumed their strongest local extrema over scales. In this sense, scale normalization from the detection scales \hat{t} constitutes a general mechanism for establishing a *common scale-invariant reference frame* with regard to scaling transformations.

It should be noted, however, that multiple extrema over scales may in general be found in the scale-space signature, as can be seen in Figs. 1 and 2, where two significant local extrema over scales are obtained in each scale-space signature, with the coarser-scale response corresponding to the lamp as a whole and the finer-scale response corresponding to the light bulb inside. Because of this inherent multi-scale nature of real-world objects, a vision system intended to interpret images from a natural environment must in general be able to handle *multiple scale hypotheses over scales*.

Scale-Space Extrema

The notion of scale selection from scale-normalized derivatives can be complemented by *spatial selection* by detecting points in scale space that assume *local extrema with respect to both space x and scale t* . Such points are referred to as *scale-space extrema*. Specifically, detection of scale-space extrema of rotationally invariant differential invariants provides a general, effective, and robust methodology for detecting *interest points with built-in scale selection* [10, 11, 15].

Thus, given a scale-normalized differential expression $\mathcal{D}_{\gamma-norm} L$, one simultaneously obtains spatial positions \hat{x} and scale estimates \hat{t} according to

$$(\hat{x}; \hat{t}) = \underset{(x; t)}{\operatorname{argmaxminlocal}} (\mathcal{D}_{\gamma-norm} L)(x; t). \quad (13)$$

Figure 3 shows the result of detecting the 50 strongest scale-space extrema of the scale-normalized Laplacian $\nabla_{norm}^2 L$ and the scale-normalized determinant of the Hessian $\det \mathcal{H}_{norm} L$ from an image that contains two objects of different sizes. Each scale-space extremum has been illustrated by a circle with the radius proportional to the detection scale $\hat{\sigma} = \sqrt{\hat{t}}$. In Fig. 4, each feature has been visualized by a sphere in the 3-D scale-space volume of the 2-D image, with the radius of the sphere increasing with the detection scale. As can be seen from this illustration, the notion of scale-space extrema can effectively reveal interest points and characteristic scales of those (see the “Applications” section below for more details about scale-invariant interest point detectors). Specifically, the differences in the radii of the circles in the 2-D illustration and in the heights over the image plane in the 3-D graphics reveal the scale differences between corresponding image features from the two objects.

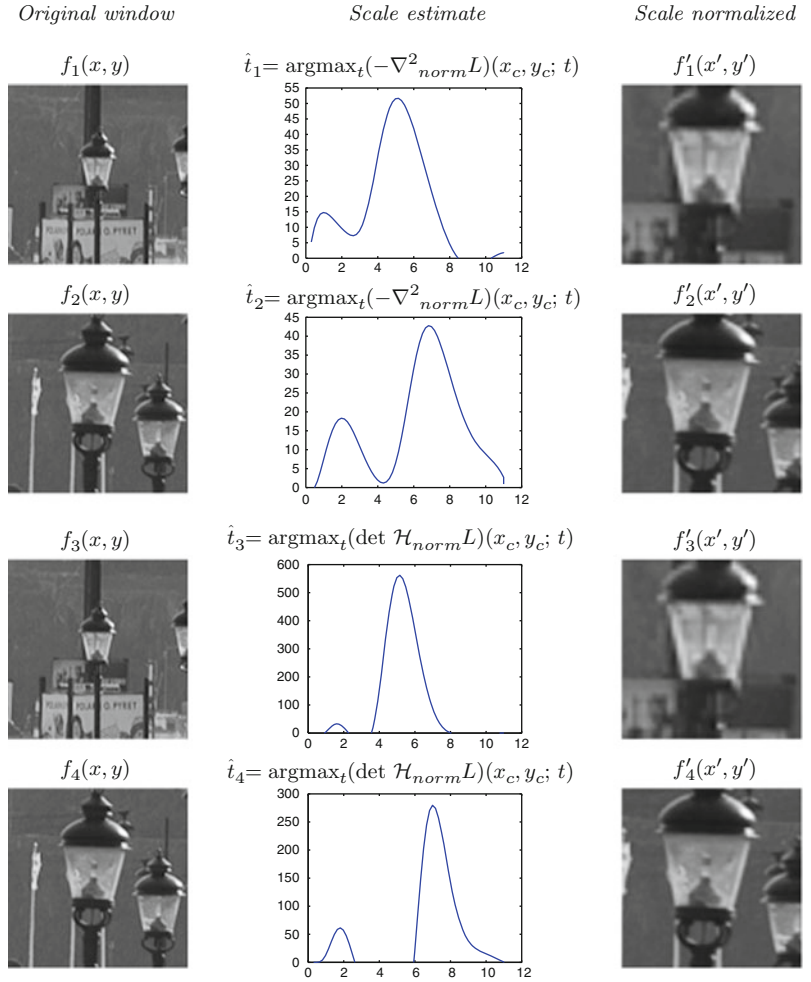
The differential operators $\nabla_{norm}^2 L$ and $\det \mathcal{H}_{norm} L$ in general both produce strong responses at the centers of blob-like structures that are either brighter or darker than their surrounding, provided that these differential entities are computed at scale levels that roughly match the size of the corresponding image structures. For this reason, they constitute very useful differential entities for *blob detection*.

Discrete Implementation

Detection of scale-space extrema from an N -dimensional discrete image can be performed by nearest-neighbor comparisons in the $N + 1$ -dimensional scale-space volume. For a 2-D image, this implies that nearest-neighbor comparisons are performed by local comparisons with the 26 neighbors in a $3 \times 3 \times 3$ neighborhood over space and scale [10] [11, footnote 16] [20]. Scale estimates and position estimates of higher accuracy can then be obtained by fitting a parabola to the data around any scale-space extremum [17, 20].

Discrete analogues of γ -normalized derivatives can be obtained either by (i) *variance-based normalization*

Scale Selection, Fig. 2 In this illustration, local windows around two of the lamps in Fig. 1 (shown in the *left column*) have been rescaled by scaling factors $\hat{\sigma} = \sqrt{\hat{t}}$ obtained from the dominant response over scales of the Laplacian $\nabla_{norm}^2 L$ and the determinant of the Hessian $\det \mathcal{H}_{norm} L$ (shown in the *middle column*) to compute a *scale-normalized window* (shown in the *right column*) around each detail. In this way, scale selection can be used for defining a *scale-normalized reference frame* for subsequent computation of *scale-invariant image descriptors* (Reprinted from [27] with permission)



which implies that the discrete derivative approximations $\delta_{x^\alpha} L$ are multiplied by an appropriate power of the scale parameter

$$L_{\xi^\alpha}(\cdot; t) = L_{\xi_1^{\alpha_1} \dots \xi_N^{\alpha_N}}(\cdot; t) = (t^{\gamma/2})^{|\alpha|} (\delta_{x^\alpha} L)(\cdot; t) \quad (14)$$

or by using the notion of (ii) l_p -normalization [11, Appendix A.4.2]:

$$L_{\xi^\alpha}(\cdot; t) = L_{\xi_1^{\alpha_1} \dots \xi_N^{\alpha_N}}(\cdot; t) = C_\alpha (\delta_{x^\alpha} L)(\cdot; t), \quad (15)$$

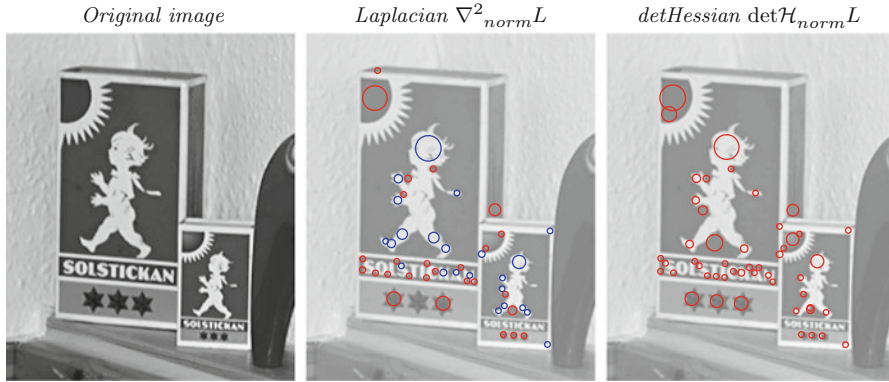
where the discrete normalization constants C_α are determined such that the l_p -norms of the scale-normalized discrete derivative approximation kernels $\delta_{x^\alpha} T$ [10, Chap. 5] are to be equal to the L_p -norms of the corresponding γ -normalized Gaussian derivative kernels $\partial_{x^\alpha} g$

$$C_\alpha \left(\sum_{n \in \mathbb{R}^N} |(\delta_{x^\alpha} T)(n; t)|^p \right)^{1/p} = (t^{\gamma/2})^{|\alpha|} \left(\int_{x \in \mathbb{R}^N} |(\partial_{x^\alpha} g)(x; t)|^p dx \right)^{1/p}. \quad (16)$$

Experiments in [17] show that the notion of l_p -normalization gives more accurate scale estimates in situations where discretization effects become important.

A particularly convenient way of implementing scale-space smoothing in this context is by convolution with the *discrete analogue of the Gaussian kernel* [10, pp 84–87]:

$$T(n; t) = e^{-t} I_n(t), \quad (17)$$



Scale Selection, Fig. 3 2-D illustration of the 50 strongest *scale-space extrema* of the Laplacian $\nabla^2_{norm} L$ and the determinant of the Hessian $\det \mathcal{H}_{norm} L$ computed from an image with two similar objects of different physical sizes. Each feature is

illustrated by a circle centered at the position (\hat{x}, \hat{y}) of the scale-space extremum and with the radius proportional to the detection scale $\hat{\sigma} = \sqrt{\hat{i}}$. *Red circles* represent scale-space maxima, while *blue circles* represent scale-space minima

which implies that semigroup property of the Gaussian scale space holds exactly also for the discrete scale-space kernels $T(\cdot; t_1) * T(\cdot; t_2) = T(\cdot; t_1 + t_2)$ and the *cascade smoothing property*

$$L(\cdot; t_2) = T(\cdot; t_2 - t_1) * L(\cdot; t_1), \quad (18)$$

for $t_2 \geq t_1 \geq 0$ implies that one can perform a set of incremental convolutions with kernels of smaller support instead of computing each scale level from the original signal f independently.

The notion of scale selection from scale-normalized derivatives can also be transferred to a *pyramid* representation to allow for real-time implementation on standard processors [3, 17, 20].

Alternative Approaches to Scale Selection

A number of other mechanisms for scale selection have also been developed based on ideas of

- Detecting peaks over scales in weighted entropy measures [7] or Lyapunov functionals [25]
- Minimizing normalized error measures over scales in order to compute more accurate localization estimates for coarser-scale corner features [11, Sect. 7.2] or for coarse-to-fine matching of highly noisy image data [13]
- Determining minimum reliable scales for feature detection according to an a priori determined noise suppression model [4]
- Determining optimal stopping times in nonlinear diffusion-based image restoration methods using

similarity measurements relative to the original data [22]

- Performing image segmentation from the scales at which a supervised classifier delivers class labels with the highest posterior [18, 19]

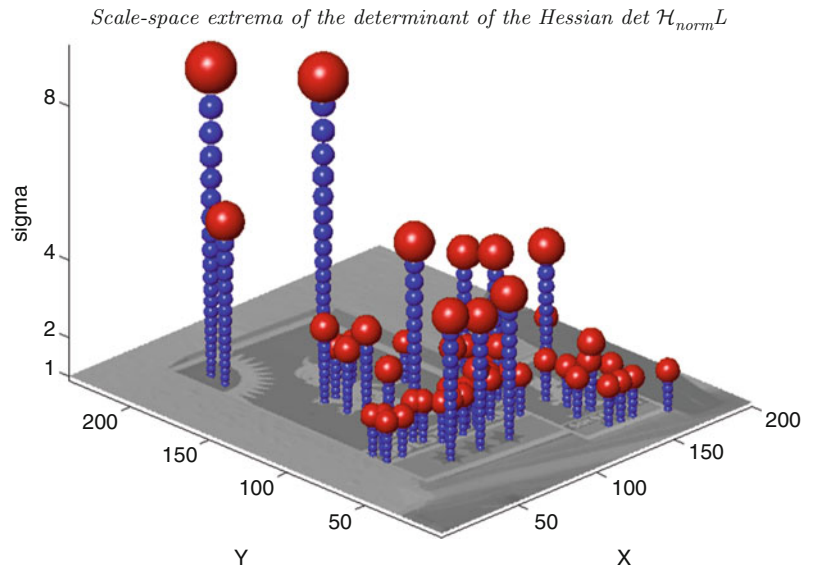
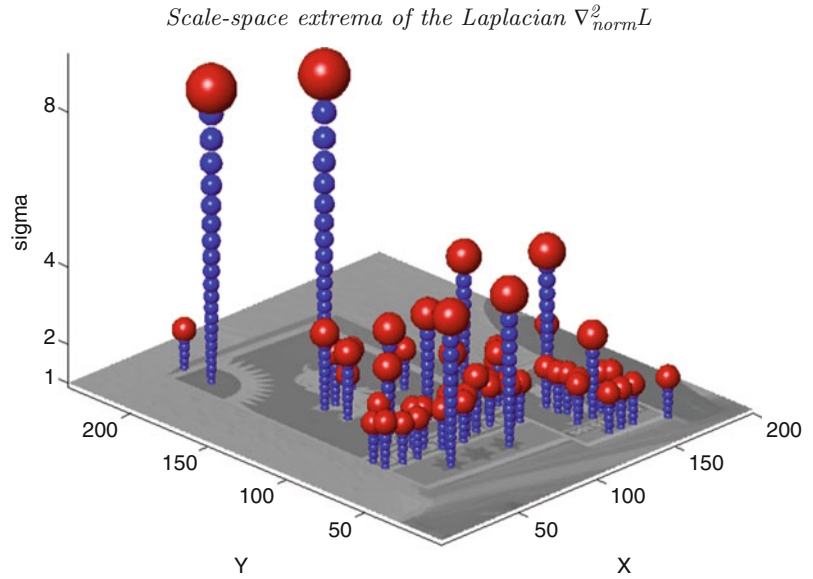
Relations Between the Different Approaches to Scale Selection

The different approaches to scale selection may have quite different properties, depending on the types of data they are applied to. For noise-free data, an adaptive noise suppression scheme optimized for suppressing high-frequency noise can be expected to not smooth the data at all, thus implying the selection of a zero scale, whereas scale selection based on local extrema over scales will always select a scale level reflecting a characteristic length in the image data.

Provided that the characteristic lengths of the relevant image features are greater than the typical characteristic lengths in the noise, scale selection based on scale-normalized derivatives will therefore lead to scale-invariant image features. Smoothing approaches that are optimized for suppressing superimposed high-frequency noise will on the other hand lead to an amount of smoothing that is primarily determined by the noise level and therefore not necessarily corresponding to scale-invariant image descriptors. In this respect, these two types of scale determination approaches can lead to fundamentally different results.

If the task is to detect fine-scale details with amplitude and/or characteristic scales comparable to the

Scale Selection, Fig. 4 3-D illustration of the 50 strongest *scale-space extrema* of the Laplacian $\nabla_{norm}^2 L$ and the determinant of the Hessian $\det \mathcal{H}_{norm} L$ computed from the image in Fig. 4. Here, each feature is illustrated by a *red sphere* centered at the position $(\hat{x}_0, \hat{y}_0; \hat{t}_0)$ of the scale-space extremum and with the radius increasing with the detection scale \hat{t}_0 . The *blue spheres* have been inserted to simplify visual interpretation



noise, it does, however, not seem unlikely that the two types of approaches could possibly benefit from each other.

Application

Interest Point Detectors with Built-in Scale Selection

Below, four different interest point detectors with automatic scale selection will be presented. A more general set of scale-invariant interest point detectors defined

according to a similar methodology can be found in [15] with an in-depth theoretical analysis of their scale selection properties in [16].

Blob Detection

Based on the notion of scale-space extrema, straightforward methods for blob detection can be obtained by detecting scale-space extrema of either (i) the scale-normalized Laplacian $\nabla_{norm}^2 L = t(L_{xx} + L_{yy})$ or (ii) the scale-normalized determinant of the Hessian $\det \mathcal{H}_{norm} L = t^2(L_{xx}L_{yy} - L_{xy}^2)$ [10, 11]. Specifically, using the *Laplacian* operator one can detect

- Bright blobs from negative scale-space minima of $\nabla_{norm}^2 L$
- Dark blobs from positive scale-space maxima of $\nabla_{norm}^2 L$

Using *the determinant of the Hessian*, one can on the other hand detect

- Bright blobs from positive scale-space maxima of $\det \mathcal{H}_{norm} L$ that satisfy $\nabla^2 L < 0$
- Dark blobs from positive scale-space maxima of $\det \mathcal{H}_{norm} L$ that satisfy $\nabla^2 L > 0$
- Saddle-like image features from negative scale-space minima of $\det \mathcal{H}_{norm} L$

These two blob detection approaches do both satisfy the basic scale selection property that if the scale-adaptive blob detector is applied to a two-dimensional Gaussian blob with scale value t_0 , i.e., $f(x, y) = g(x, y; t_0)$, then the select scale \hat{t} will be equal to the scale of the blob in the input data, i.e., $\hat{t} = t_0$.

In comparison, the image features obtained from the determinant of the Hessian blob detector do often have better repeatability properties under affine image deformations than Laplacian image features [11, 15, 16].

Figure 5 shows the result of applying these interest point detectors to a grey-level image. Please note how the variations in the detection scales of the blob responses reflect the perspective scaling effects in the scene.

Corner Detection

A straightforward method for scale-invariant corner detection can be obtained by detecting positive scale-space maxima and negative scale-space minima of the scale-normalized *rescaled level curve curvature* measure

$$\tilde{\kappa}(L) = t^{2\gamma} |\nabla L|^2 \kappa(L) = t^{2\gamma} (L_x^2 L_{yy} + L_y^2 L_{xx} - 2L_x L_y L_{xy}), \quad (19)$$

where $\kappa(L)$ denotes the curvature of the level curves of the Gaussian smoothed image at any scale and $\gamma = 7/8$ turns out to be a good choice [11, Sect. 6] [15]; see Fig. 6a for an illustration.

The *Harris-Laplace* operator [21] is structurally different in the respect that it uses different entities for spatial selection (the Harris measure) and scale selection ($\nabla_{norm}^2 L$); see Fig. 6b.

Edge Detection

With regard to edge detection, the evolution properties over scales of the *scale-normalized gradient magnitude*

$$|\nabla L|_{norm} = t^{\gamma/2} \sqrt{L_x^2 + L_y^2} \quad (20)$$

can be shown to reveal local characteristics of the type of edge [12, Sect. 4]. Specifically, by choosing $\gamma = 1/2$, a local maximum over scales will be assumed at a scale corresponding to the diffuseness of a one-dimensional diffuse step edge

$$\Phi(x; t_0) = \int_{u=-\infty}^x g(u; t_0) du \quad (21)$$

and may then provide cues to, e.g., focus blur, shadow edges, or rounded edges.

Ridge and Valley Detection

Let e_p and e_q denote the eigendirections of the Hessian matrix $\mathcal{H}L$ such that the mixed second-order derivative in this coordinate frame is zero $L_{pq} = 0$ and denote the eigenvalues of the Hessian matrix by L_{pp} and L_{qq} . These eigenvalues are also referred to as *principal curvatures*, and these directions are assumed to be ordered such that $L_{pp} < L_{qq}$.

Then, a differential geometric definition of the *ridges* in the image at any scale can be expressed as the set of points that satisfy [12, Sect. 5.2]

$$L_p = 0, \quad L_{pp} \leq 0, \quad |L_{pp}| \geq |L_{qq}|. \quad (22)$$

Similarly, the *valleys* at any scale can be defined from [14]

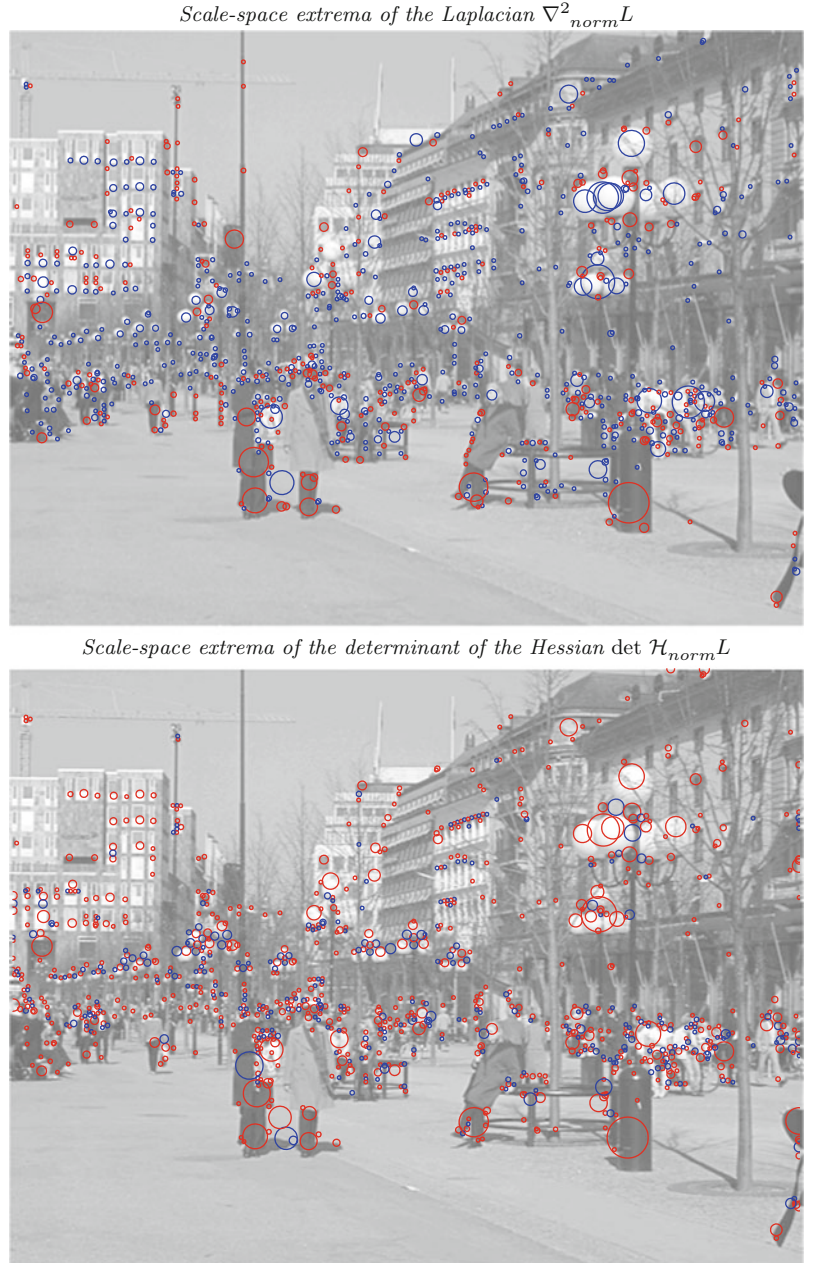
$$L_q = 0, \quad L_{qq} \geq 0, \quad |L_{qq}| \geq |L_{pp}|. \quad (23)$$

With $R_{\gamma-norm}$ denoting a scale-normalized measure of *ridge strength* (or valley strength) defined from the principal curvatures L_{pp} and L_{qq} , one can also express ridge and valley detection methods with automatic scale selection by detecting *scale-space ridges* using the definition

$$L_p = 0, \quad L_{pp} \leq 0, \quad \partial_t(R_{\gamma-norm}) = 0, \quad \partial_{tt}(R_{\gamma-norm}) \leq 0 \quad (24)$$

Scale Selection, Fig. 5

Scale-invariant interest points obtained from the 1,000 strongest scale-space extrema of the Laplacian $\nabla^2_{norm} L$ and the determinant of the Hessian $\det \mathcal{H}_{norm} L$ with the size of each circle reflecting the detection scale of the corresponding feature. *Red circles* represent local maxima of the operator response, while *blue circles* indicate local minima (Adapted from [15])



and *scale-space valleys* according to

$$\begin{aligned} L_q &= 0, & L_{qq} &\geq 0, & \partial_t(R_{\gamma-norm}) &= 0, \\ \partial_{tt}(R_{\gamma-norm}) &\leq 0. \end{aligned} \quad (25)$$

Specifically, it can be shown that for natural measures of ridge or valley strength, the choice $\gamma = 3/4$ implies

that the selected scale will reflect the width of a Gaussian ridge (or valley) [12]. For generalizations to 3-D images, see [6, 9, 24].

Feature Tracking

By adapting the scales for feature detection by a local scale selection mechanism, the resulting image features will be robust to scale changes, which means that they can be matched over substantial

Scale Selection, Fig. 6

Scale-invariant interest points obtained from the 1,000 strongest scale-space extrema of the rescaled level curve curvature $\tilde{\kappa}(L)$ and the Harris-Laplace operator. The size of each circle reflects the detection scale of the corresponding feature. For the rescaled level curve curvature operator $\tilde{\kappa}(L)$, the color of the circles show the sign of the curvature; *red circles* represent a local maxima of the operator response, while *blue circles* indicate local minima (Adapted from [15])

Scale-space extrema of the rescaled level curve curvature $\tilde{\kappa}(L)$



The Harris-Laplace operator



size variations [2]. Indeed, the variations over time in the characteristic scale estimates obtained during feature tracking can, if appropriately implemented, be robust enough for computing estimates of *time to collision* [17].

Image-Based Matching and Recognition

The SIFT descriptor [20] comprises a bottom-up keypoint detection stage with scale-space extrema

detection in a differences-of-Gaussians (DoG) pyramid. The scale-invariant properties of the SIFT descriptor can be explained as follows.

From the way that the DoG operator is implemented in the pyramid in [20], it follows that the normalization will be similar to the scale-normalized Laplacian. Using the fact that the scale-space representation satisfies the diffusion equation, it follows that the Laplacian operator can be approximated from the difference between two levels in the scale-space representation:

$$\begin{aligned}
\frac{1}{2}\nabla^2 L(x, y; t) &= \partial_t L(x, y; t) \\
&\approx \frac{L(x, y; t + \Delta t) - L(x, y; t)}{\Delta t} \\
&= \frac{DOG(x, y; t, \Delta t)}{\Delta t}, \quad (26)
\end{aligned}$$

i.e., from the difference of two Gaussian smoothed images.

With the scale levels distributed such that the ratio between successive scale levels is k when measured in terms of $\sigma = \sqrt{t}$, (i.e., $\sigma_{i+1} = k \sigma_i$ and $t_{i+1} = k^2 t_i$ which implies that $\Delta t_i = (k^2 - 1) t_i$), it follows that [15]

$$\begin{aligned}
DOG(x, y; t) &= L(x, y; k^2 t) - L(x, y; t) \\
&\approx (k^2 - 1) t (\partial_t L(x, y; t)) = (k^2 - 1) t \frac{1}{2} \nabla^2 \\
L(x, y; t) &= \frac{(k^2 - 1)}{2} t \nabla^2 L(x, y; t) \\
&= \frac{(k^2 - 1)}{2} \nabla_{norm}^2 L(x, y; t). \quad (27)
\end{aligned}$$

Hence, with self-similar sampling of the scale levels, the pyramid-implemented DoG interest point operator can be interpreted as an approximation of the scale-adapted Laplacian operator in Eq. (11).

In the SURF descriptor [1], local feature detection is performed by detecting local extrema over space and scale of an approximation of the determinant of the Hessian operator in terms of Haar wavelets, with the filters normalized to constant l_1 - or Frobenius norm over scales. According to Eq. (9), the γ -normalized derivative concept corresponds to normalization of the Gaussian derivative operators to unit L_p -norm over scales. Furthermore, it was shown in [17] that normalizing the filter responses to constant l_p -norm over scales gives better accuracy in a practical implementation than normalization of the discrete filters by multiplication with the scale parameter raised to a power of $m\gamma/2$, where m denotes the order of differentiation. Hence, the initial feature detection step in the SURF descriptor can be seen as an approximation of the scale-normalized determinant of the Hessian operator in Eq. (12).

The scale-invariant property of the actual image descriptors in the SIFT and SURF descriptors does in turn follow from the scale-invariant properties of

the initial feature detection step, in line with the general framework for computing scale-invariant image descriptors from scale estimates obtained from local extrema over scales of scale-normalized differential expressions, as described in the “Theory” section above.

In these ways, the notion of scale selection constitutes a general mechanism for computing scale-invariant image descriptors for image-based matching and recognition.

References

1. Bay H, Ess A, Tuytelaars T, van Gool (2008) Speeded up robust features (SURF). *Comput Vis Image Underst* 110(3):346–359
2. Bretzner L, Lindeberg T (1998) Feature tracking with automatic selection of spatial scales. *Comput Vis Image Underst* 71(3):385–392
3. Crowley J, Riff O (2003) Fast computation of scale normalised receptive fields. In: *Proceedings of scale-space’03. Lecture Notes in Computer Science*, vol 2695. Springer, Berlin/New York, pp 584–598
4. Elder JH, Zucker SW (1998) Local scale control for edge detection and blur estimation. *IEEE Trans Pattern Anal Mach Intell* 20(7):699–716
5. Field DJ (1987) Relations between the statistics of natural images and the response properties of cortical cells. *J Opt Soc Am* 4:2379–2394
6. Frangi AF, Niessen WJ, Hoogeveen RM, vanWalsum T, Viergever MA (2000) Model-based quantitation of 3D magnetic resonance angiographic images. *IEEE Trans Med Imaging* 18(10):946–956
7. Kadir T, Brady M (2001) Saliency, scale and image description. *Int J Comput Vis* 45(2):83–105
8. Koenderink JJ (1984) The structure of images. *Biol Cybern* 50:363–370
9. Krissian K, Malandain G, Ayache N, Vaillant R, Troussset Y (2000) Model-based detection of tubular structures in 3D images. *Comput Vis Image Underst* 80(2):130–171
10. Lindeberg T (1994) Scale-space theory in computer vision. *The Springer international series in engineering and computer science*. Springer, Berlin/New York
11. Lindeberg T (1998) Feature detection with automatic scale selection. *Int J Comput Vis* 30(2):77–116
12. Lindeberg T (1998) Edge detection and ridge detection with automatic scale selection. *Int J Comput Vis* 30(2):117–154
13. Lindeberg T (1998) A scale selection principle for estimating image deformations. *Image Vis Comput* 16(14):961–977
14. Lindeberg T (2008) Scale-space. In: Wah B (ed.) *Encyclopedia of computer science and engineering*. Wiley, pp 2495–2504. [dx.doi.org/10.1002/9780470050118.ecse609](https://doi.org/10.1002/9780470050118.ecse609). Also available from <http://www.nada.kth.se/~tony/abstracts/Lin08-EncCompSci.html>
15. Lindeberg T (2010) Interest point from scale-space features: scale-space primal sketch for differential descriptors *International Journal of Computer Vision*

16. Lindeberg T (2013) Scale selection properties of generalized scale-space interest point detectors. *J Mathematical Imaging and Vision*, 46(2):177–210
17. Lindeberg T, Bretzner L (2003) Real-time scale selection in hybrid multi-scale representations. In: *Proceedings of Scale-Space Methods in Computer Vision. Lecture Notes in Computer Science*, vol 2695. Springer, Berlin/New York, pp 148–163
18. Loog M, Li Y, Tax D (2009) Maximum membership scale selection. In: *Multiple Classifier Systems. Lecture Notes in Computer Science*, vol 5519. Springer, Berlin, pp 468–477.
19. Li Y, Tax DMJ, Loog M (2012) Supervised scale-invariant segmentation (and detection). In: *Proc. Scale Space and Variational Methods in Computer Vision (Scale-Space'11)*, Ein Gedi, Israel. *Lecture Notes in Computer Science*, vol 6667. Springer, Berlin, pp 350–361
20. Lowe D (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
21. Mikolajczyk K, Schmid C (2004) Scale and affine invariant interest point detectors. *Int J Comput Vis* 60(1):63–86
22. Mrázek P, Navara M (2003) Selection of optimal stopping time for nonlinear diffusion filtering. *Int J Comput Vis* 52(2–3):189–203
23. Negre A, Brailion C, Crowley JL, Laugier C (2008) Real-time time-to-collision from variation of intrinsic scale. *Exp Robot* 39:75–84
24. Sato Y, Nakajima S, Shiraga N, Atsumi H, Yoshida S, Koller T, Gerig G, Kikinis R (1998) 3D multi-scale line filter for segmentation and visualization of curvilinear structures in medical images. *Med Image Anal* 2(2):143–168
25. Sporring J, Colios, CJ, Trahanias, PE (2000) Generalized scale-selection. In: *Proc. International Conference on Image Processing (ICIP'00)*, Vancouver, Canada, pp 920–923
26. Witkin AP (1983) Scale-space filtering. In: *Proceedings of 8th International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany, pp 1019–1022
27. Lindeberg T (2013) Invariance of visual operations at the level of receptive fields. *PLoS ONE* 8(7): e66990. doi:10.1371/journal.pone.0066990

Scale-Invariant Image Features and Image Descriptors

► [Scale Selection](#)

Scene/Image Parsing

► [Semantic Image Segmentation](#)

Schott Noise

► [Photon, Poisson Noise](#)

Second Order Approximation

► [Osculating Paraboloids](#)

Segmentation

► [Geodesics, Distance Maps, and Curve Evolution](#)
 ► [Interactive Segmentation](#)

Semantic Image Segmentation

Jamie Shotton¹ and Pushmeet Kohli²

¹Microsoft Research Ltd, Cambridge, UK

²Department of Computer Science And Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel

Synonyms

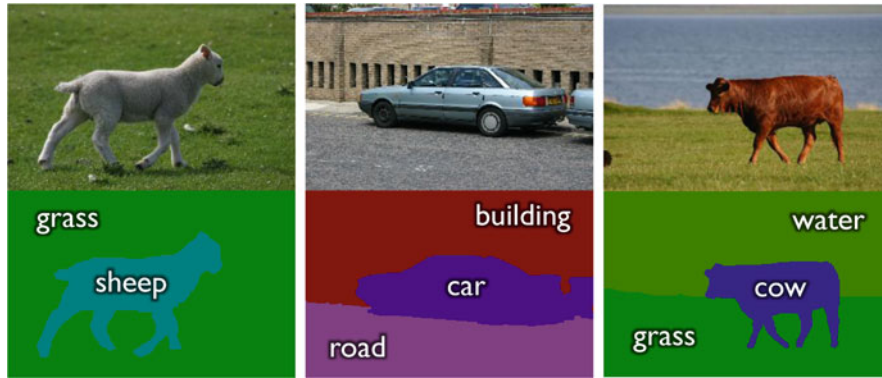
[Object segmentation](#); [Scene/image parsing](#)

Definition

Semantic image segmentation describes the task of partitioning an image into regions that delineate meaningful objects and labeling those regions with an object category label. Some example semantic segmentations are given in [Fig. 1](#). It can be seen as a generalization of figure-ground segmentation [1] where one segments a particular object, say a horse, from the background.

Background

Images typically contain multiple objects, including *things* such as people, cars, and cows and *stuff* such as grass, sky, and water. The imaging process composites the appearances of these objects, leaving (at most) an intensity edge between one object and the next. Semantic image segmentation aims to recover the image regions corresponding directly to objects, as well as labeling those regions with the relevant object category.



Semantic Image Segmentation, Fig. 1 Top row: three input images. Bottom row: the corresponding semantic segmentations where colors represent object categories

The task is usually approached as supervised or semi-supervised machine learning, using a set of training images that are manually segmented and labeled. The learning algorithm then discovers relevant image features that help discriminate regions belonging to different categories in unseen test images. Other cues such as layout and context further help resolve ambiguities; for example, a pixel that neighbors a car pixel is likely to have the same label, and a region containing a cow is likely to be above a region containing grass.

Semantic image segmentation has been of interest since at least 1989 [2], but only fairly recently [3, 4] have processor speeds started to allow the rich models for high accuracy across many object categories.

Theory

Problem Formulations In essence, the semantic segmentation task can be treated as a pixel-labeling problem. The different methods proposed for solving this problem can be categorized based on the relationships they encode between different pixels (see Fig. 2). Some methods for semantic segmentation solve the pixel-labeling problem by classifying each pixel independently [5, 6]. Another class of methods works by grouping pixels into segments (or super-pixels) and assigning a single label to each group [7]. Super-pixels are computed from the image in a bottom-up fashion [8–10] and can aid computational efficiency but may lead to a final incorrect labeling.

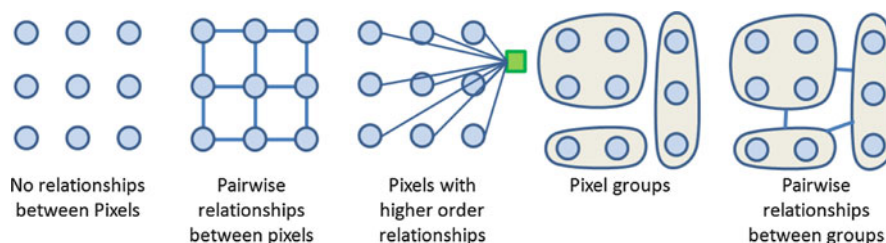
Many semantic segmentation algorithms are based on the pairwise Markov Random Field (MRF) [11]

or Conditional Random Field (CRF) models [12, 13] which enforce relationships between pairs of neighboring pixels [5]. They encourage adjacent pixels that are similar in appearance to take the same semantic label, and lead to segmentation results with smooth boundaries. A related notion of layout consistency was explored in [14].

Some methods go beyond pairwise interactions between pixels and enforce higher-order relationships between groups of (or even all) pixels in the image. For instance, they encourage groups of pixels to take the same semantic label [15], or make sure that some semantic label is taken by at least one pixel in the image [16]. These models also allow the use of top-down object detection results to prime the segmentation [6, 17]. In related work, data-driven Markov Chain Monte Carlo (DDMCMC) was used to parse images with a rich generative model [18]. A nonparametric approach to semantic segmentation was proposed in [19], where labels were transferred between nearest-neighbor images matched using SIFT flow features.

Features Used for Pixel/Super-Pixel Classification

There are many informative image cues that can be used for semantic segmentation, including intensity, color, texture, context, motion, and 3D structure. Dense interest point descriptors can be used, for example, [20–22], or light-weight region integrals of textons [5, 6], or even features based on histogram of gradients [16]. Motion-derived 3D structure was used in [23] to segment images from a video sequence. Range images or depth cameras have been used for semantic segmentation in [24].



Semantic Image Segmentation, Fig. 2 Different approaches for semantic image segmentation

Incorporating Context

Context can be incorporated in several ways: (i) *appearance* context [5] captures the notion that a sheep might typically stand on something green, (ii) *semantic* context [6, 25–27] captures the notion that a sheep might typically stand on grass, and (iii) *auto-context* [6, 28] describes a *recursive* classification procedure whereby the classification uses contextual features from a previous stage of classification, capturing the notion that this pixel might belong to sheep because a nearby pixel was classified as sheep at the previous stage.

Datasets

To a large extent, the quality of the semantic segmentation algorithm is related to the quality of the training dataset. Most techniques require fully pixel-wise labeled images which are expensive to obtain. Other approaches reduce this requirement in various ways, including multiple over-segmentations [29], co-segmenting several images at once [30], incorporating image or region labels [6, 17, 31], and exploiting probabilistic aspect models [32].

Early datasets for semantic segmentation include Corel and Sowerby [33] and the MSRC dataset [5]. Recently, more challenging dataset have been proposed that dramatically increase the variability in the images and thus bring us much closer to solving semantic segmentation as a real-world problem. LabelMe [34] used a web interface to capture a large number of image labels. The Pascal VOC Segmentation Challenge [35] runs yearly and deliberately tries to remove contextual clues in the data to foster the best object detection algorithms. The SUN 09 dataset [36] goes to the other extreme of labeling many objects per image to include as much context as possible. LabelMe, Pascal VOC, and SUN 09 for the most part only provide labels for things and not stuff,

often leaving the background a single heterogeneous category.

Application

Accurately segmenting and recognizing what those objects are opens up many potential applications. Not only does it tell us what things are in an image but also where they are and how they look. This information can be used in, for example, image search, image editing, augmented reality, robot navigation, and medical image analysis.

References

1. Borenstein E, Ullman S (2002) Class-specific, top-down segmentation. In: Heyden A, Sparr G, Johansen P (eds) Proceedings of the European conference on computer vision. Lecture notes in computer science (ECCV), vol 2351. Springer, Berlin, pp 109–124
2. Wright W (1989) Image labelling with a neural network. In: Proceedings of the 5th Alvey vision conference, Reading
3. Everingham M, Thomas B, Troscianko T (1999) Head-mounted mobility aid for low vision using scene classification techniques. *Int J Virtual Real* 3(4):3–12
4. Konishi S, Yuille AL (2000) Statistical cues for domain specific image segmentation with performance analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Hilton Head, vol 1, pp 125–132
5. Shotton J, Winn J, Rother C, Criminisi A (2009) Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int J Comput Vis* 81(1):2–23
6. Shotton J, Johnson M, Cipolla R (2008) Semantic texton forests for image categorization and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Anchorage
7. Campbell N, Mackeown W, Thomas B, Troscianko T (1997) Interpreting image databases by region classification. *Pattern Recognit* 30:555–563

8. Shi J, Malik J (1997) Normalized cuts and image segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, San Juan, pp 731–737
9. Felzenszwalb P, Huttenlocher D (2004) Efficient graph-based image segmentation. *Int J Comput Vis* 59(2): 167–181
10. Carreira J, Sminchisescu C (2010) Constrained parametric min-cuts for automatic object segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, San Francisco
11. Geman S, Geman D (1984) Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans Pattern Anal Mach Intell* 6(6):721–741
12. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: *Proceedings of the international conference on machine learning*, Williams College, pp 282–289
13. Kumar S, Hebert M (2003) Discriminative random fields: a discriminative framework for contextual interaction in classification. In: *Proceedings of the international conference on computer vision, Kerkyra*, vol 2, pp 1150–1157
14. Winn J, Shotton J (2006) The layout consistent random field for recognizing and segmenting partially occluded objects. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Miami, vol 1, pp 37–44
15. Kohli P, Ladický L, Torr P (2009) Robust higher order potentials for enforcing label consistency. *Int J Comput Vis* 82:302–324
16. Ladický L, Russell C, Kohli P, Torr P (2010) Graph cut based inference with co-occurrence statistics. In: *Proceedings of the European conference on computer vision (ECCV)*, Heraklion
17. Ladický L, Sturges P, Alahari K, Russell C, Torr P (2010) What, where and how many? Combining object detectors and CRFs. In: *Proceedings of the European conference on computer vision (ECCV)*, Heraklion
18. Tu Z, Chen X, Yuille A, Zhu S (2003) Image parsing: unifying segmentation, detection, and recognition. In: *Proceedings of the IEEE conference on computer vision, Nice, France*, vol 1, pp 18–25
19. Liu C, Yuen J, Torralba A (2009) Nonparametric scene parsing: label transfer via dense scene alignment. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Miami
20. Lowe D (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
21. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. *IEEE Trans Pattern Anal Mach Intell* 24(4):509–522
22. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, San Diego, vol 2, pp 886–893
23. Brostow G, Shotton J, Fauqueur J, Cipolla R (2008) Segmentation and recognition using structure from motion point clouds. In: *Proceedings of the European conference on computer vision (ECCV)*, Marseille
24. Anguelov D, Taskar B, Chatalbashev V, Koller D, Gupta D, Ng A (2005) Discriminative learning of markov random fields for segmentation of 3D scan data. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Beijing
25. Torralba A, Murphy K, Freeman W, Rubin M (2003) Context-based vision system for place and object recognition. In: *Proceedings of the conference on computer vision, Nice, France*, vol 2, pp 273–280
26. Hoiem D, Efros A (2006) Objects in perspective. In: *Proceedings of the conference on computer vision and pattern recognition (CVPR)*, New York
27. Rabinovich A, Vedaldi A, Galleguillos C, Wiewiora E, Belongie S (2007) Objects in context. In: *Proceedings of the international conference on computer vision, Rio de Janeiro*
28. Tu Z (2008) Auto-context and its application to high-level vision tasks. In: *Proceedings of the international conference on computer vision and pattern recognition (CVPR)*, Anchorage
29. Russell BC, Efros AA, Sivic J, Freeman WT, Zisserman A (2006) Using multiple segmentations to discover objects and their extent in image collections. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, New York
30. Rother C, Minka T, Blake A, Kolmogorov V (2006) Cosegmentation of image pairs by histogram matching – incorporating a global constraint into MRFs. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, New York
31. Barnard K, Duygulu P, de Freitas N, Forsyth D, Blei D, Jordan MI (2003) Matching words and pictures. *J Mach Learn Res* 3:1107–1135
32. Verbeek J, Triggs B (2007) Region classification with markov field aspect models. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Minneapolis
33. He X, Zemel R, Carreira-Perpiñán M (2004) Multiscale conditional random fields for image labeling. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, Washington, DC, vol 2, pp 695–702
34. Russell B, Torralba A, Murphy K, Freeman WT (2008) Labelme: a database and web-based tool for image annotation. *Int J Comput Vis* 77:157–173
35. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A, The PASCAL VOC challenge. <http://www.pascal-network.org/challenges/VOC/>
36. Choi M, Lim J, Torralba A, Willsky A (2010) Exploiting hierarchical context on a large database of object categories. In: *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, San Francisco

Semiautomatic

► Interactive Segmentation

Semidefinite Optimization

► Semidefinite Programming

Semidefinite Programming

Chunhua Shen and Anton van den Hengel
School of Computer Science, The University of
Adelaide, Adelaide, SA, Australia

Synonyms

Convex minimization; Semidefinite optimization

Definition

Semidefinite programming is a subtopic of convex optimization. Convex optimization refers to minimization of a convex function subject to a set of convex constraints. Semidefinite programming involves minimization of a linear objective function over the intersection of linear constraints and the cone of positive semidefinite matrices. Clearly, semidefinite programming is a special case of convex optimization.

Background

Many computer vision problems can be formulated as convex optimization problems. The main advantage of convex optimization is that if a local minimum exists, then it is also a global minimum. In other words, the convexity guarantees to attain the global optimum if it exists.

In a semidefinite programming problem, one minimizes a linear function subject to the constraint that an affine combination of symmetric matrices is positive semidefinite. Semidefinite programming unifies a few standard problems such as linear programming, quadratic programming, and second-order cone programming. Semidefinite programming has many applications in computer vision.

Theory

Mathematically, semidefinite programming solves the following problem:

$$\begin{aligned} \min_X \langle C, X \rangle, \text{ s.t. } \langle A_k, X \rangle = b_k, (k = 1, \dots, m), \\ X \succeq 0. \end{aligned} \quad (1)$$

Here, the optimization variable $X \in \mathcal{R}^{D \times D}$ is a symmetric and positive semidefinite matrix. The operator $\langle A, B \rangle = \sum_{ij} A_{ij} B_{ij}$ calculates the inner product of two matrices (or vectors). The last constraint $X \succeq 0$ means X is positive semidefinite. Such a constraint is nonlinear and nonsmooth but convex. The following statements about a semidefinite matrix are equivalent: (1) $X \succeq 0$; (2) All eigenvalues of X are nonnegative; and (3) $\forall \mathbf{u} \in \mathcal{R}^D, \mathbf{u}^\top X \mathbf{u} \geq 0$.

We can easily write the dual problem of (Eq. 1) by finding the saddle point of its Lagrangian:

$$\max_{\mathbf{y}} \langle \mathbf{b}, \mathbf{y} \rangle, \text{ s.t. } \sum_{k=1}^m y_k A_k \preceq C. \quad (2)$$

Here, $\mathbf{y} \in \mathcal{R}^m$ is the variable to optimize. The notation $A \preceq B$ means $B - A \succeq 0$. Weak duality and strong duality hold for the primal problem (Eq. 1) and dual problem (Eq. 2) under mild conditions. In particular, strong duality means that optimal values of the primal and dual problems are the same. Strong duality follows from Slater's condition for constraint qualification [1], which is stated as follows. If (Eq. 1) and (Eq. 2) both are feasible and there is a strictly interior point for either (Eq. 1) or (Eq. 2), then optimal primal and dual solutions exist and the corresponding optimal values are the same.

Semidefinite programming can be viewed as an extension of linear programming where the element-wise inequalities between vectors become linear matrix inequalities (LMIs). An LMI takes the form of $\sum_{k=0}^m y_k A_k \succeq 0$, with $\mathbf{y} = [y_0 \dots y_m]$ being a vector and matrices A_k being symmetric matrices. An LMI enforces a convex constraint on the vector \mathbf{y} . Indeed, the interior-point methods for linear programming can be generalized to semidefinite programming. In fact, as shown by Nesterov and Nemirovsky [2], in principle interior-point methods for linear programming can be generalized to all convex optimization problems. Significantly, interior-point methods enable semidefinite programs to be solved very efficiently. It is theoretically guaranteed that the effort needed to solve a semidefinite program to a prescribed accuracy grows no faster than a polynomial of the problem size. Off-the-shelf solvers include CSDP [3], SDPT3 [4], and SeDuMi [5]. One can also use optimization modeling languages such as CVX [6] and YALMIP [7] to greatly simplify the problem modeling process.

Although interior-point methods are polynomial-time algorithms, they do not scale well. If the number of constraints m is of order $O(D^2)$ with the semidefinite matrix being $D \times D$, at each iteration, the computational time is of order $O(D^6)$. This high computational complexity makes interior-point methods impractical for large-scale problems. Recently, first-order augmented Lagrangian approaches have been proposed for solving large problems. In particular, as shown in [8], the alternating direction method is applied to solve the dual augmented Lagrangian problem of the standard semidefinite problem. This method is much more scalable than the conventional interior-point methods.

Application

Semidefinite programming has attracted extensive research interests in computer vision and machine learning due to its many applications in these fields.

Max-cut approximation and image segmentation. It is well known that semidefinite programming can be used to approximately solve some difficult combinatorial optimization problems. It is a useful tool for approximating NP-hard problems. Such an example is the Max-cut problem, which can be described as follows. Given a graph $G = (V, E)$, output a partition of the vertices V so as to maximize the number of edges crossing from one side to the other. This problem can be expressed as an integer quadratic problem:

$$\max_{\mathbf{u}} \sum_{(i,j) \in E} \frac{1 - u_i u_j}{2}, \text{ s.t. } u_i \in \{-1, 1\}, \forall i = 1, \dots, n. \quad (3)$$

This problem is NP-hard due to the discrete constraints. To obtain a semidefinite relaxation, we lift each scalar variable u_i to a higher dimension $\mathbf{u}_i \in \mathcal{R}^n$ and $\|\mathbf{u}\|_2 = 1$. We introduce another variable substitution $U_{ij} = \mathbf{u}_i^\top \mathbf{u}_j$ to bring (Eq. 3) into a semidefinite program:

$$\max_U \sum_{(i,j) \in E} \frac{1 - U_{ij}}{2}, \text{ s.t. } U_{ii} = 1, \forall i = 1, \dots, n; U \succeq 0. \quad (4)$$

The constraint $U \succeq 0$ is due to the fact $U_{ij} = \mathbf{u}_i^\top \mathbf{u}_j$. The optimal value of (Eq. 4) is an upper bound of the

original problem (Eq. 3). It is easy to see that (Eq. 4) is a semidefinite problem. Hence, the global solution of (Eq. 4) is guaranteed. Approximate solutions to the original problem can be computed by rounding the vector solution of (Eq. 4). The solutions of the SDP formulation suggest to cut nodes i and j if $U_{ij} = \mathbf{u}_i^\top \mathbf{u}_j$ is close to -1 . The Goemans-Williamson randomized rounding technique is to choose a uniformly random hyperplane through the origin and use it to cut vectors into separate parts. See [9] for details. Similar ideas can be applied to solve combinatorial problems of minimizing quadratic functionals in binary decision variables subject to linear constraints, which has many applications in image segmentation, perceptual grouping, and image restoration. See, for example, [10].

Maximum variance unfolding. Another application is embedding high-dimensional data (such as image data) into an underlying low-dimensional space by maximizing the variance while maintaining the data's local neighborhood. Here, the input data are assumed to be sampled from a much lower dimensional manifold that is embedded inside of a higher dimensional vector space. The primary idea of maximum variance unfolding is to create a mapping that preserves local neighborhoods at every point of the underlying manifold.

Let \mathbf{u}_i ($i = 1, \dots, n$) be the original data points in the high-dimensional manifold and \mathbf{v}_i be the *unfolded* low-dimensional data points. For $(i, j) \in E$ are neighbors, the local isometry constraints are

$$\|\mathbf{v}_i - \mathbf{v}_j\|_2^2 = \|\mathbf{u}_i - \mathbf{u}_j\|_2^2 = d_{ij}, \forall (i, j) \in E. \quad (5)$$

If we constrain the embedded data to center at the origin, we have

$$\sum_i \mathbf{v}_i = \mathbf{0}. \quad (6)$$

The variance of the embedded data can be written as $\frac{1}{n} \sum_i \mathbf{v}_i^\top \mathbf{v}_i$. So the optimization problem becomes

$$\max_{\mathbf{v}} \sum_i \mathbf{v}_i^\top \mathbf{v}_i, \text{ s.t. } \sum_i \mathbf{v}_i = \mathbf{0}, \|\mathbf{v}_i - \mathbf{v}_j\|_2^2 = d_{ij}, \forall (i, j) \in E. \quad (7)$$

This problem is a non-convex quadratic program. To formulate it into a semidefinite problem, we

introduce a new variable $V_{ij} = \mathbf{v}_i^\top \mathbf{v}_j$, the same as in the first example. We can then relax (Eq. 7) into

$$\begin{aligned} \max_V \langle V, \mathbf{I} \rangle, \text{ s.t. } \sum_{ij} V_{ij} = 0, V_{ii} + V_{jj} - 2V_{ij} \\ = d_{ij}, \forall (i, j) \in E; V \succeq 0. \end{aligned} \quad (8)$$

Here, \mathbf{I} is an identity matrix. Problem (Eq. 8) is a relaxation of (Eq. 7). The optimal value of (Eq. 8) is an upper bound of the one of (Eq. 7) because the feasibility set of (Eq. 8) is a superset of (Eq. 7)'s feasibility set. The advantage of this relaxation is that now (Eq. 8) can be efficiently solved using off-the-shelf semidefinite programming solvers. The embedded points \mathbf{v} can be obtained via Cholesky decomposition of the matrix V . Weinberger and Saul [11] applied maximum variance unfolding to detect low-dimensional structure in high-dimensional data sets of images.

References

1. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, Cambridge
2. Nesterov Y, Nemirovsky A (1988) A general approach to polynomial time algorithms design for convex programming. Technical report, USSR Acad Sci, Moscow, USSR
3. Borchers B (1999) CSDP, a C library for semidefinite programming. Optim Methods Softw 11(1):613–623
4. Toh K, Todd M, Tutuncu R (1999) SDPT3—a matlab software package for semidefinite programming. Optim Methods Softw 11(1–4):545–581
5. Sturm JF (1999) Using SeDuMi 1.02, a matlab toolbox for optimization over symmetric cones. Optim Methods Softw 11(1–4):625–653
6. Grant M, Boyd S (2011) CVX: matlab software for disciplined convex programming, version 1.21. <http://cvxr.com/>
7. Löfberg J (2004) YALMIP: a toolbox for modeling and optimization in MATLAB. Proceedings of the IEEE symposium on computer-aided control system design, Taipei, Taiwan
8. Wen Z, Goldfarb D, Yin W (2009) Alternating direction augmented lagrangian methods for semidefinite programming. Math Program Comput 2(3–4):203–230
9. Goemans MX, Williamson DP (1994) .879-approximation algorithms for max cut and max 2SAT. Proceedings of the ACM symposium on theory of computing. ACM, New York, pp 422–431
10. Keuchel J, Schnörr C, Schellewald C, Cremers D (2003) Binary partitioning, perceptual grouping, and restoration with semidefinite programming. IEEE Trans. Pattern Anal Mach Intell 25(11):1364–1379
11. Weinberger KQ, Saul LK (2006) Unsupervised learning of image manifolds by semidefinite programming. Int J Comput Vision 70(1):77–90

Sensor Fusion

Pramod K. Varshney

Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY, USA

Synonyms

Multisensor data fusion

Related Concepts

► Data Fusion

Definition

Sensor fusion refers to systems, techniques, theory, and tools that exploit the synergy in the information acquired from multiple sensors to enhance system performance.

Background

Conventional systems used single sensors for monitoring phenomenon of interest and make inferences regarding them. Due to significant advances in sensing, networking, and computing technologies, multiple sensors are increasingly being used. This provides improved system performance, resulting in a better understanding of the phenomenon being monitored. In addition, distributed sensing improves robustness, extends spatial and temporal coverage while requiring shorter response time [1–3]. In order to optimally fuse information acquired from different distributed sensing architectures, advances in theory and algorithm design are required.

Theory

Data from multiple sensors can be combined at three possible levels. In data level fusion, raw sensor data is combined. This requires that data acquired from different sensors be commensurate and the data needs to be transported to a fusion center for centralized

processing. This approach has the potential of achieving the best possible performance at the expense of large communication requirements. For noncommensurate data, either feature level fusion or decision level fusion is employed. In feature level fusion, features are extracted from the data which are then fused. In decision level fusion, higher level decisions such as detections and estimates are obtained based on data from individual sensors. These decisions are then fused at the fusion center. In feature level and decision level fusion, data transmission requirements are lower, but the quality of fused result degrades due to data compression involved in the feature extraction and decision-making processes. Several topologies for sensor fusion such as parallel, serial, tree, and network topologies can be used. Choice of topology is often application dependent, but parallel topology is used quite commonly.

Sensor fusion is employed to solve a number of generic problems that results in improved situational awareness for the phenomenon under observation. Object and event detection using multisensor data is carried out based on distributed detection theory and decision fusion [4–6]. For conditionally independent observations, a likelihood ratio-based quantizer is employed at the sensors, and the fusion rule is based on a weighted sum of incoming quantized data. For parameter estimation or tracking problems, quantized data are fused at the fusion center [7, 8]. For tracking, a number of distributed filtering and track fusion algorithms are employed [9, 10]. When tracking multiple objects, data association is a major challenge [11, 12]. When sensor data is image and video data, data-level image fusion techniques are employed. One popular approach is to transform image data into another domain, e.g., wavelet domain, perform fusion in the transformed domain, and then transform the fused data back to the image domain [13]. Sensor fusion approaches are often problem and sensing modality dependent. New approaches are constantly being devised, such as distributed inference based on probabilistic graphical models [14] or through consensus and gossip algorithms [15, 16].

Application

There are many military and nonmilitary applications of sensor fusion. In military applications, sensor

fusion is employed for the detection, location, tracking, and identification of military entities such as aircrafts, ships, submarines, ground units, emitters, and weapons. Different sensing modalities such as radar, sonar, electro-optic imagers, infrared imagers, and electronic intelligence are employed. Nonmilitary applications are numerous and continue to increase. Wireless sensor networks [17, 18] and distributed camera networks [19, 20] are being deployed for many application domains. These include air traffic control, homeland security, medical diagnosis, smart homes and buildings, monitoring of critical infrastructures, robotics, vehicle health management, remote sensing, and environmental monitoring.

Open Problems

There are many theoretical and practical challenges to fully utilize the potential of distributed sensing and sensor fusion. These include scaling, fundamental limits on achievable performance, fusion of heterogeneous sensors, treatment of dependent data, and fusion of hard and soft data.

References

1. Varshney PK (1997) Scanning the issue. *Proc IEEE* 85(1): 3–5
2. Varshney PK (1997) Multisensor data fusion. *Electron Commun Eng J* 9(12):245–253
3. Liggins ME, Llinas J, Hall DL (2008) *Handbook of multisensor data fusion: theory and practice*, 2nd edn. CRC, Boca Raton
4. Varshney PK (1997) *Distributed detection and data fusion*. Springer, New York
5. Chen B, Tong L, Varshney PK (2006) Channel-aware distributed detection in wireless sensor networks. *IEEE Signal Process Mag* 23(4):16–26
6. Chamberland J-F, Veeravalli VV (2007) Wireless sensors in distributed detection applications. *IEEE Signal Process Mag* 24(3):16–25
7. Xiao J-J, Ribeiro A, Giannakis GB, Luo Z-Q (2006) Distributed compression estimation using wireless sensor networks. *IEEE Signal Process Mag* 23(4): 27–41
8. Ribeiro A, Schizas I, Roulmliotis S, Giannakis G (2010) Kalman filtering in wireless sensor networks. *IEEE Control Syst Mag* 30(2):66–86
9. Bar-Shalom Y, Blair WD (2000) *Multitarget-multisensor tracking: applications and advances*, vol III. Artech House, Boston

10. Mallick M, Krishnamurthy V, Vo B-N (2011) Integrated tracking, classification, and sensor management: theory and applications. Wiley/IEEE
11. Mahler Ronald PS (2007) Statistical multisource-multitarget information fusion. Artech House, Boston
12. Bar-Shalom Y, Daum F, J Huang J (2009) The probabilistic data association filter. *IEEE Control Syst Mag* 29(6): 82–100
13. Blum R, Liu Z (2005) Multi-sensor image fusion and its applications. CRC, Boca Raton
14. Çetin M, Chen L, Fisher III JW, Ihler AT, Moses RL, Wainwright MJ, Willsky AS (2006) Distributed fusion in sensor networks: a graphical models perspective. *IEEE Signal Process Mag* 23(4):42–55
15. Olfati-Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. *Proc IEEE* 95(1):215–233
16. Dimakis AG, Kar S, Moura JMF, Rabbat MG, Scaglione A (2010) Gossip algorithms for distributed signal processing. *Proc IEEE* 98(11):1847–1864
17. Akyldiz IF, Vuran MC (2010) Wireless sensor networks. Wiley, Chichester/Hoboken
18. Swami A, Zhao Q, Hong Y-W, Tong L (2007) Wireless sensor networks: signal processing and communications perspectives. Wiley, Chichester/England/Hoboken
19. Foresti GL, Regazzoni CS, Varshney PK (2003) Multisensor surveillance systems: the fusion perspective. Kluwer, Boston
20. Bhanu B, Ravishankar CV, Roy-Chowdhury AK, Aghajan H, Terzopoulos D (eds) (2011) Distributed video sensor networks. Springer, London/New York

Shadow

► [Penumbra and Umbra](#)

Shape from Outlines of Projection

► [Shape from Silhouette](#)

Shape from Scatter

Mohit Gupta
Department of Computer Science, Columbia
University, New York, NY, USA

Synonyms

[Depth from Scattering](#)

Definition

Light scatters in the presence of volumetric media such as fog, smoke, mist, dust, and murky water. Volumetric scattering results in several daily life visual effects, such as the glow around the streetlights and car headlights on a foggy day and the murky appearance of underwater scenes. From a computer vision point of view, while on one hand, volumetric scattering degrades images by reducing contrast, it also provides important shape/depth cues, especially for outdoor scenes. This entry provides a summary of various techniques and algorithms for recovering shape/depth using scattering.

Background

Volumetric scattering in the atmosphere (atmospheric scattering) has been studied for over two centuries in the atmospheric optics literature. Some of the prominent sources of literature on the subject are books by Minnaert [1], Middleton [2], and McCartney [3]. In computer vision literature, Cozman and Krotkov [4] and Narasimhan, Schechner, and Nayar [5–12] were among the first to develop techniques for scene analysis under volumetric scattering. Excerpts from these papers are used in several locations in this entry. Most of the algorithms presented in these papers required capturing two or more images of the scene under different weather/imaging conditions. Note that while the models and techniques in this entry are discussed in the context of atmospheric scattering, they are valid in general for other volumetric scattering scenarios, such as underwater imaging [13, 14].

Theory

Mechanisms of Atmospheric Scattering: There are two main mechanisms for getting depth from atmospheric scattering: attenuation and airlight.

Attenuation: Light gets deflected and absorbed as it travels through a volumetric medium (e.g., haze, mist, murky water), resulting in exponential *attenuation* of the intensity. Consider a collimated beam of light traveling through a medium from point A to point B. Suppose the irradiance of the beam at point A is

$E_A(\lambda)$, where λ is the wavelength of light. The irradiance of the beam at point B after the attenuation is given as

$$E_B(\lambda) = E_A(\lambda) e^{-\int_A^B \sigma(x, \lambda) dx}, \quad (1)$$

where $\sigma(x, \lambda)$ is the attenuation coefficient of the medium for light of wavelength λ . $\sigma(x, \lambda)$ is directly proportional to the volume density $\rho(x)$ of the medium. For homogeneous media, $\sigma(x, \lambda)$ is constant with respect to the spatial location, that is, $\sigma(x, \lambda) = \sigma(\lambda)$. Thus, in the presence of homogeneous media, the attenuated intensity at point B is given as

$$E_B(\lambda) = E_A(\lambda) e^{-\sigma(\lambda)d}, \quad (2)$$

where d is the distance between points A and B.

So far in this entry, light attenuation only for a collimated light beam has been considered. For diverging beams from point light sources, there is an additional inverse-square falloff. Given a point light source at point A with a radiant intensity of $I_o(\lambda)$, the irradiance at point B is

$$E_B(\lambda) = g \frac{I_o(\lambda) e^{-\sigma(\lambda)d}}{d^2}, \quad (3)$$

where the constant g accounts for the optical parameters of the camera and conversion between radiant intensity and irradiance.

Airlight: A second mechanism causes the atmosphere to behave like a source of light. This phenomenon is called airlight [15], and it is caused by the scattering of environmental illumination by particles in the atmosphere towards the observer. The environmental illumination can have several sources, including, direct sunlight, diffuse skylight, and light reflected by the ground. Consider an observer at a distance d from a physical object. The total radiance due to airlight at the observer is given as (for a derivation, see [5])

$$L(d, \lambda) = L(\infty, \lambda) (1 - e^{-\sigma(\lambda)d}), \quad (4)$$

where $L(\infty, \lambda)$ is the radiance due to airlight for an object at infinity (e.g., horizon). While attenuation causes the scene radiance to decrease with path length, airlight increases with path length. It therefore causes the apparent brightness of a scene point to increase with depth.

Application

1. *Depth from Attenuation:* Consider the image of a scene at night. Suppose the scene has artificial light sources, for example, street lights, building windows, and car headlights. There is no airlight due to sunlight or skylight. In this setting, the dominant scattering mechanism is attenuation. The observed irradiance due to a light source of radiance intensity $I(\lambda)$ at a distance d is

$$E(d, \lambda) = g \frac{I(\lambda) e^{-\sigma(\lambda)d}}{d^2}, \quad (5)$$

where, as before, $\sigma(\lambda)$ is the attenuation coefficient of the medium and g is the parameter which accounts for the conversion between radiant intensity and irradiance. If the detector of the camera has spectral response $s(\lambda)$, the final image brightness value recorded is determined as

$$\begin{aligned} E'(d) &= \int s(\lambda) E(d, \lambda) d\lambda \\ &= \int g s(\lambda) \frac{I(\lambda) e^{-\sigma(\lambda)d}}{d^2} d\lambda. \end{aligned} \quad (6)$$

Since the spectral bandwidth of the camera is limited (visible light range when camera is black and white and even narrower spectral bands when the camera is color), it is safe to assume the attenuation coefficient $\sigma(\lambda)$ to be constant over this bandwidth. Then

$$E'(d) = g \frac{e^{-\sigma d}}{d^2} \int s(\lambda) I(\lambda) d\lambda = g \frac{e^{-\sigma d}}{d^2} I'. \quad (7)$$

If the light source is observed under two different weather conditions, for example, dense fog and mild fog (or one condition could be clear weather with $\sigma = 0$), there are two different attenuation coefficients σ_1 and σ_2 . Taking the ratio of the two resulting image brightness values,

$$R = \frac{E'_1}{E'_2} = e^{-(\sigma_1 - \sigma_2)d}. \quad (8)$$

Using the natural log,

$$R' = \ln R = (\sigma_1 - \sigma_2)d. \quad (9)$$

This quantity is independent of the sensor gain and the radiant intensity of the source. By computing the above quantity (log of ratio) for two different light sources in the scene and taking their ratio, the relative depths of the two source locations can be computed:

$$\frac{R'_i}{R'_j} = \frac{d_i}{d_j} . \quad (10)$$

Hence, the relative depths of all sources (with unknown radiant intensities) in the scene can be computed from two images taken under unknown but different haze or fog conditions.

2. *Depth from Airlight*: Under dense fog and close-by objects or mild fog and distant objects, attenuation of object brightness is severe, and airlight is the main cause of image irradiance. Also, in the case of dense haze around noon, airlight dominates. In these scenarios, scaled scene depths can be computed from a single image [5, 9].

Let a scene point at depth d produce airlight radiance $L(d, \lambda)$. If the camera has a spectral response $s(\lambda)$, the final brightness value recorded for the scene point is

$$E'(d) = g \int s(\lambda) L(d, \lambda) d\lambda , \quad (11)$$

where, as before, g accounts for the constant of proportionality between scene radiance and image irradiance. Substituting the model of airlight from Eq. 4:

$$E'(d) = \int g s(\lambda) L(\infty, \lambda) (1 - e^{-\sigma(\lambda)d}) d\lambda . \quad (12)$$

By assuming that the scattering coefficient $\sigma(\lambda)$ is more or less constant over the spectral band of the camera,

$$E'(d) = E(\infty) (1 - e^{-\sigma d}) . \quad (13)$$

This equation gives a relationship between the observed airlight intensity E' and the scaled scene depth σd . $E(\infty)$ is the image intensity corresponding to a scene point at infinity and can be measured from the image if the horizon is visible (this part can be identified as the brightest region of the image). Then, the scaled depth can be computed as

$$\sigma d = \ln \left(\frac{E(\infty)}{E(\infty) - E'(d)} \right) . \quad (14)$$

3. *Depth from Chromatic Decomposition*: So far in the entry, attenuation and airlight have been considered separately. At night, there can be no airlight (since there is no environmental illumination) and hence, attenuation dominates. In contrast, under dense fog or haze during daylight, the radiance from a scene point is severely attenuated and hence airlight dominates. However, in most situations, the effects of both attenuation and airlight coexist.

Narasimhan and Nayar presented a technique [6, 9] to recover depths for these situations by considering the chromatic effects of atmospheric scattering. The key idea is that the spectral composition of the irradiance at a scene point is the weighted sum of two distributions, corresponding to the direct transmission after attenuation and airlight. The weights are a function of the weather conditions. Using this model, the authors showed that it is possible to recover depths by capturing and performing simple arithmetic operations on two images of the scene under different bad weather conditions. For details, the reader is referred to Refs. [6, 9].

4. *Depth from Dehazing*: There are several techniques which explicitly remove the effects of scattering from the images, for example, using polarization [8, 10] or image priors [16–18]. While the primary goal of these *dehazing* techniques is to improve visibility in the images, as a by-product, scene depths are also recovered by using the technique outlined in the section *Depth from Airlight* on the separated haze layer (airlight).

Open Problems

A long-standing open problem has been *single image dehazing*, that is, removing the effects of atmospheric scattering using *a single image*. This would enable image and scene recovery for dynamic scenes. Recently, there have been a few techniques which have addressed this problem [16–18]. In order to account for the under-constrained nature of the problem, these techniques use a variety of scene priors [16–18].

References

1. Minnaert M (1954) The nature of light and color in the open air. Dover, New York

2. Middleton WEK (1952) Vision through the atmosphere. University of Toronto Press, Toronto
3. McCartney EJ (1976) Optics of the atmosphere: scattering by molecules and particles. Wiley, New York
4. Cozman F, Krotkov E (1997) Depth from scattering. Proc IEEE conf on Computer Vision and Pattern Recognition (CVPR), San Juan. 801–806
5. Nayar SK, Narasimhan SG (1999) Vision in bad weather. Proc IEEE ICCV 2:820–827
6. Narasimhan SG, Nayar SK (2000) Chromatic framework for vision in bad weather. Proc IEEE CVPR 1: 598–605
7. Narasimhan SG, Nayar SK (2001) Removing weather effects from monochrome images. Proc IEEE CVPR 2:186–193
8. Schechner YY, Narasimhan SG, Nayar SK (2001) Instant dehazing of images using polarization. Proc IEEE CVPR 1:325–332
9. Narasimhan SG, Nayar SK (2002) Vision and the atmosphere. IJCV 48(3):233–254
10. Schechner YY, Narasimhan SG, Nayar SK (2003) Polarization-based vision through haze. Applied Optics 42(3):511–525
11. Narasimhan SG, Nayar SK (2003) Shedding light on the weather. Proc IEEE CVPR 1:665–672
12. Narasimhan SG, Nayar SK (2003) Interactive deweathering of an image using physical models. In: IEEE Workshop on Color and Photometric Methods in Computer Vision, In Conjunction with ICCV, Nice, France
13. Treibitz T, Schechner Y (2006) Instant 3descatter. Proc IEEE conf Computer Vision and Pattern Recognition (CVPR), New York, NY, pp 1861–1868
14. Schechner YY, Karpel N (2005) Recovery of underwater visibility and structure by polarization analysis. IEEE J Ocean Eng 30(3):570–587
15. Koschmieder H (1924) Theorie der horizontalen sichtweite. Beitr Phys freien Atm 12:33–53, 171–181
16. Tan RT (2008) Visibility in bad weather from a single image. Proc IEEE conf on Computer Vision and Pattern Recognition (CVPR), Anchorage, AK. 801–806
17. Fattal R (2008) Single image dehazing. ACM Trans Graphics 27(3):72:1–9
18. He K, Sun J, Tang X (2009) Single image haze removal using dark channel prior. Proc IEEE Trans Pattern Analysis and Machine Intelligence. 33(12):2342–2353

Shape from Shadows

Pascal Mamassian

Laboratoire Psychologie de la Perception, Université Paris Descartes, Paris, France

Definition

Shape from shadows is the inference of the three-dimensional shape of objects from their shadows.

Background

David Waltz first introduced to computer vision the problem of shape from shadows while attempting to segment objects that had sharp edges in a three-dimensional scene under various lighting conditions [1]. The problem was then refined to arbitrary surfaces [2]. Nowadays, shape from shadows is a cue commonly used when one is interested in inferring the three-dimensional structure of a visual scene. A benefit relative to other cues such as stereopsis or motion parallax is that it is available in a single image. Shape from shadows differs from shape from shading because in the latter the useful information is the gradient of reflected light on a curved surface, whereas shape from shadows considers that the information coming from the area inside cast shadow is uniform (there are no variations inside the shadow that are related to the scene geometry [3]). However, it is important to remember that similar to most cues to three-dimensional shape, shape from shadows is an under-constrained problem in the sense that several critical scene parameters are often unknown, in particular the light source position and the nature of the surface on which the shadow is cast.

Shadow Identification

The first serious problem in shape from shadows is to identify a region of the image as a shadow. Dark regions on a surface can be caused by the absence of illumination or by a darker surface reflectance (albedo). The cues that allow humans to distinguish variations in light intensity from variations in material properties include luminance relations across shadow borders, figural relations, 3D-shape, depth, color, texture, and motion [4]. In computer vision, very promising results have been obtained if a shading model is assumed [5].

Another dichotomy exists between self-shadows (or attached shadows, i.e., shadows cast on the object itself) and cast shadows (shadows cast on a remote surface) [6]. This differentiation can be achieved to some extent by using some invariant color properties [7]. In addition, a variety of geometrical constraints of self-shadows on a smooth surface have been derived [8, 9]. The list of shadow properties that human and artificial visual systems could use is still open [10].

Shadow Correspondence Problem

In order for shadows to be useful, appropriate parts of an object should be matched to corresponding parts of shadows. The choice of the appropriate features to solve the correspondence problem is still a topic of intense research [11]. Potential candidates include points of high curvature along the object and shadow contours, some Fourier components for textured surfaces [12], or only the low spatial frequencies if one is merely interested in inferring where an object is relative to the background [11]. The shadow correspondence problem is more complex when the receiving surface is not flat. In particular, perceptual problems are thought to arise when the receiving surface is saddle shaped [13].

Using multiple images of the scene under various illuminations, especially if the light source positions are controlled in an active way, is highly beneficial [14, 15]. Multiple images can also be useful to detect the presence of concavities in an object [16].

Open Problems

Object segmentation and recognition are believed to be easier when shadows are absent from the image [17]. Is it true, and if so, what is the best method to eliminate shadows in a scene?

The dual problem of shape from shadows is to infer the illumination from shadows [18]. Should we separately estimate the shape and the illumination from shadows or is it better to attempt to infer both simultaneously?

Moving shadows bring new challenges for object segmentation and tracking [19]. So why do human observers appear to excel at extracting the spatial layout of a scene when moving shadows are present [20]?

References

1. Waltz DL (1975) Understanding line drawings of scenes with shadows. In: Winston PH (ed) *The psychology of computer vision*. McGraw-Hill, New York, pp 19–91
2. Shafer SA, Kanade T (1983) Using shadows in finding surface orientations. *Comput Vis Graph Image Process* 22:145–176
3. Cavanagh P, Leclerc YG (1989) Shape from shadows. *J Exp Psychol* 15(1):3–27

4. Kingdom FAA (2008) Perceiving light versus material. *Vis Res* 48(20):2090–2105
5. Bell M, Freeman WT (2001) Learning local evidence for shading and reflectance. In: *Proceedings of international conference on computer vision (ICCV)*, Vancouver, BC, vol 1, pp 670–677
6. Mamassian P, Knill DC, Kersten D (1998) The perception of cast shadows. *Trends Cogn Sci* 2(8):288–295
7. Salvador E, Green P, Ebrahimi T (2001) Shadow identification and classification using invariant color models. In: *Proceedings of ICASSP 01*, Salt Lake City, UT, vol 3 (IEEE), pp 1545–1548
8. Knill DC, Mamassian P, Kersten D (1997) Geometry of shadows. *J Opt Soc Am A* 14(12):3216–3232
9. Hatzitheodorou M (1998) Shape from shadows: a Hilbert space setting. *J Complex* 14(1):63–84
10. Casati R (2004) The shadow knows: a primer on the informational structure of cast shadows. *Percept* 33(11):1385–1396
11. Mamassian P (2004) Impossible shadows and the shadow correspondence problem. *Percept* 33(11):1279–1290
12. Ramamoorthi R, Koudelka M, Belhumeur P (2005) A Fourier theory for cast shadows. *IEEE Trans Pattern Anal Mach Intell* 27:288–295
13. Norman JF, Lee YL, Phillips F, Norman HF, Jennings LR, McBride TR (2009) The perception of 3-D shape from shadows cast onto curved surfaces. *Acta Psychol* 131(1):1–11
14. Clark JJ, Wang L (2001) Active shape-from-shadows with controlled illuminant trajectories. *Int J Comput Vis* 43(3):141–166
15. Kriegman D, Belhumeur P (2001) What shadows reveal about object structure. *J Opt Soc Am A* 18:804–1813
16. Savarese S, Andreetto M, Rushmeier H, Bernardini F, Perona P (2007) 3D reconstruction by shadow carving: theory and practical evaluation. *Int J Comput Vis* 71(3):305–336
17. Finlayson G, Hordley S, Lu C, Drew M (2006) On the removal of shadows from images. *IEEE Trans Pattern Anal Mach Intell* 28:59–68
18. Sato I, Sato Y, Ikeuchi K (2003) Illumination from shadows. *IEEE Trans Pattern Anal Mach Intell* 25:290–300
19. Jiang H, Drew MS (2007) Shadow resistant tracking using inertia constraints. *Pattern Recognit* 40(7):1929–1945
20. Kersten D, Mamassian P, Knill DC (1997) Moving cast shadows induce apparent motion in depth. *Percept* 26(2):171–192

Shape from Silhouette

David C. Schneider

Image Processing Department, Fraunhofer Heinrich Hertz Institute, Berlin, Germany

Synonyms

[Shape from outlines of projection](#); [Visual hull](#)

Related Concepts

► Visual Hull

Definition

Shape from silhouette (SfS) algorithms compute the (approximate) 3D shape of an object from multiple 2D projections considering only the outline of the object in the projections. The most important class of SfS methods are Visual Hull algorithms.

Background

The problem of computing shape from outlines (silhouettes) of projections is generally underconstrained. Depending on the object's geometry and the number of available views, approximations can be computed which are sufficient for some applications. Often, the SfS output serves as initialization for appearance-based shape reconstruction methods such as volumetric reconstruction or multiview stereo.

The most general class of SfS approaches, where no assumptions about the type of objects to reconstruct are made, are the Visual Hull algorithms (e.g., [1–4]). These are treated in a dedicated article.

If the class of objects to handle is restricted, specialized model-based algorithms can solve SfS-like problems even for a single view. In this case, the 3D shape is not actually reconstructed. The algorithms rather compute a shape within the scope of the model that satisfies silhouette constraints and otherwise maximizes domain-specific priors (e.g., a likelihood with respect to a learned distribution). Examples are [5, 6], where human body pose is estimated from a single silhouette, or [7] where the shape of a head is estimated from a face profile.

References

1. Baumgart BG (1974) Geometric modeling for computer vision. PhD thesis, Stanford, Stanford, CA, USA
2. Ahuja N, Veenstra J (1989) Generating octrees from object silhouettes in orthographic views. IEEE Trans Pattern Anal Mach Intell 11(2):137–149

3. Laurentini A (1991) The visual hull: a new tool for contour-based image understanding. In: Proceedings of the 7th Scandinavian conference on image analysis. Pattern Recognition Society of Denmark, Aalborg East
4. Szeliski R (1993) Rapid octree construction from image sequences. CVGIP 58:23–32
5. Balan AO, Sigal L, Black MJ, Davis JE, Haussecker HW (2007) Detailed human shape and pose from images. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR '07). IEEE, Piscataway, Minneapolis, Minnesota, pp 1–8
6. Guan P, Weiss A, Balan AO, Black MJ (2009) Estimating human shape and pose from a single image. In: Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, pp 1381–1388
7. Schneider DC, Eisert P (2009) Fast nonrigid mesh registration with a data-driven deformation prior. In: Proceedings of the IEEE 12th international computer vision workshops, Kyoto, Japan, pp 304–311

Shape from Specular Reflections

► Shape from Specularities

Shape from Specularities

Silvio Savarese

Department of Electrical and Computer Engineering,
University of Michigan, Ann Arbor, MI, USA

Synonyms

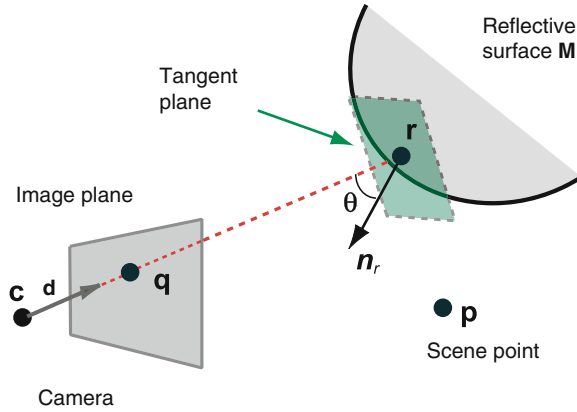
[Shape from specular reflections](#)

Definition

Specular reflections or *specularities* carry valuable information about the geometry of reflective surfaces and can be used to recover their shape.

Background

Cues such as texture and shading are often inadequate for recovering the shape of shiny reflective objects. For such objects it is not possible to observe their surfaces directly, rather only what they reflect. Yet, specular reflections present an additional cue that potentially



Shape from Specularities, Fig. 1 The basic shape from specularities problem. Given a scene point reflected off a specular surface M at \mathbf{r} and given its observation in the camera image plane at \mathbf{q} , the goal is to recover 3D location of \mathbf{r} and higher-order local parameters of M around \mathbf{r}

may be exploited for shape recovery. A curved mirror produces “distorted” images of the surrounding world. For example, the image of a straight line reflected by a curved mirror is, in general, a curve (Fig. 2). It is clear that such distortions are systematically related to the shape of the surface. Is it possible to invert this map and recover the shape of the mirror from its reflected images? The general “inverse mirror” problem is under-constrained: by opportunely manipulating the surrounding world, we may produce a great variety of images from any curved mirror surface. This inverse problem may become tractable under the assumption that some knowledge about the structure of the scene, the shape of the object, or the observer is available.

Theory

The basic shape from specularities problem is formulated as follows: Let M be a reflective surface whose shape is unknown (Fig. 1). Consider a camera observing M and let \mathbf{c} be the camera projection center. Given a scene point (or light point source) \mathbf{p} , let \mathbf{q} be the image of \mathbf{p} observed in the image plane through a specular reflection on the reflective surface at \mathbf{r} . Let \mathbf{n}_r be the unit normal to the surface at \mathbf{r} (i.e., the normal of the tangent plane of M at \mathbf{r}). The main objective of the shape from specularities problem is to recover the 3D location of \mathbf{r} in the camera reference system

given the observation \mathbf{q} . Often, it is desirable to recover higher-order local shape information around \mathbf{r} . In this case the objective is to estimate \mathbf{r} , the unit normal \mathbf{n}_r , and second- or higher-order local parameters of M at \mathbf{r} such as, for instance, directions and magnitudes of the surface principal curvature at \mathbf{r} .

If the camera is calibrated (i.e., if the internal camera parameters are known), the surface position at \mathbf{r} is completely determined by a single distance parameter s . This is easy to show: It follows from the perspective projection constraint that the point \mathbf{r} must belong to the line defined by \mathbf{c} and \mathbf{q} , resulting in the following relationship:

$$\mathbf{r} = \mathbf{c} + s \mathbf{d}, \quad (1)$$

where the unit vector $\mathbf{d} = (\mathbf{q} - \mathbf{c}) / \|\mathbf{q} - \mathbf{c}\|$ is parallel to the line of sight and $s = \|\mathbf{r} - \mathbf{c}\|$ is the distance from \mathbf{c} to \mathbf{r} . It follows that \mathbf{r} is known up to one degree of freedom (i.e., s). Notice that no information about the surface normal \mathbf{n}_r or other higher-order surface parameters can be obtained in this case.

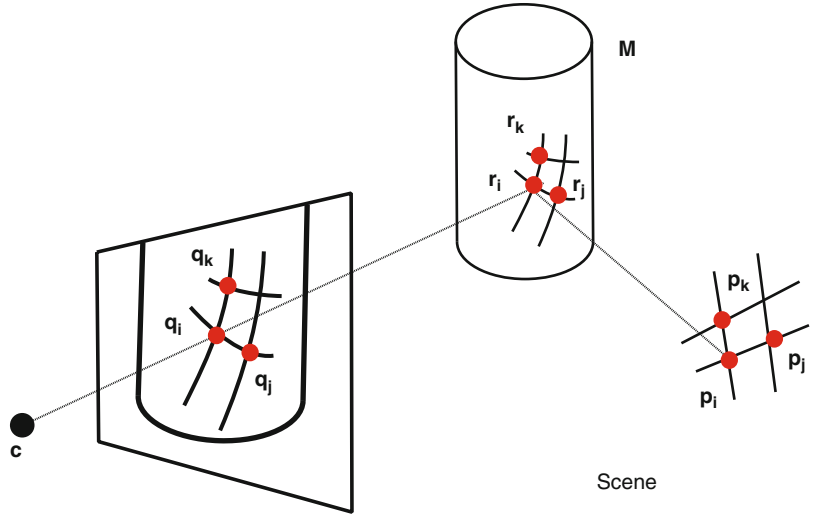
Assumptions on geometrical configuration of the scene (light source) or the reflectance properties of the surface lead to further constraints for estimating s and higher-order local shape information around \mathbf{r} . Some of the most notable cases are:

- *Single calibrated scene point (light point source).* If one assumes that the location of \mathbf{p} is known in the camera reference system, it can be shown that \mathbf{r} and \mathbf{n}_r are known up to one-dimensional family of solutions parameterized by s [1, 2]. This unknown can be determined if one further assumes that \mathbf{p} is at infinity [3].
- *Extended calibrated scene (light source).* The following assumptions are made. Assume that (at least) three points $\{\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k\}$ can be identified within a neighborhood of a scene planar patch and their reflection $\{\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k\}$ off the surface M are observable in the image plane as $\{\mathbf{q}_i, \mathbf{q}_j, \mathbf{q}_k\}$ (Fig. 2). Assume that correspondences between image points and scene points can be established (e.g., one knows that \mathbf{q}_i is the observation of reflection \mathbf{r}_i of scene point \mathbf{p}_i , for any i). Then, it can be shown that location of $\{\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k\}$ and corresponding surface normals at $\{\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k\}$ can be recovered as long as $\{\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k\}$ are close enough and are not colinear. Moreover, second-order information of M at $\{\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k\}$ (i.e., directions and magnitudes of

Shape from

Specularities, Fig. 2

Local shape of a reflective surface M can be recovered up second order if (at least) three points $\{\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k\}$ (along with their observations $\{\mathbf{q}_i, \mathbf{q}_j, \mathbf{q}_k\}$) can be identified within a small planar patch in the scene [1]



principal curvature) can be estimated up to one-dimensional family of solutions. Details of this analysis are presented in the theoretical work by Savarese et al. [1] where necessary and sufficient conditions for reconstructing the local shape of a specular surface from one or multiple reflected calibrated points and (or) lines are discussed.

- *Known reflectance model for the surface.* If the reflected light is modeled using a physical reflectance model such as the Torrance-Sparrow BRDF model [4], a reflected point on the surface can be in general characterized by an extended pattern which is typically indicated as a *highlight*. It can be shown that by measuring the radiance falloff of specular highlights [5], the local curvature can be estimated.

Several practical methods have been proposed for recovering the shape of specular surfaces that use a setup similar to that in Figs. 1 and 2 wherein a calibrated fixed camera observes a scene element (light source) being reflected off the surface. Methods vary depending on whether the scene is calibrated or uncalibrated, static or time varying, and point source or extended. Pioneering work by Ikeuchi [3] and, later, works by Sanderson et al. [6] utilize distant light sources which are sequentially activated so as to generate sequences of point source reflections. Halstead et al. [7] and Tarini et al. [8] use extended scene structures composed of conically shaped patterns and a sequence of striped patterns, respectively. These methods recover the complete surface of a shiny object by iteratively fitting a parametric representation of the

shape to local measurements. Zheng and Murata [9] propose a system where the object rotates while being illuminated by extended radial light sources. Savarese et al. [1] use a single calibrated grid pattern similar to that in Fig. 2 to obtain sparse local surface estimates (up to second order). Rozenfeld et al. [10] extend this analysis and demonstrate that dense reconstruction can be obtained from sparse correspondences between scene points and observations. Kutulakos and Steger [11] exploit directed ray measurements of a calibrated planar target reflected off the surface and positioned at different locations in the world reference system. Adato et al. [12] show that it is possible to relax the hypothesis of calibrated scene by analyzing the reflection of a time varying distant unknown scene observed by an orthographic camera.

Additional constraints for estimating the reflection point \mathbf{r} (along with higher-order shape information) can be obtained by having multiple observations of \mathbf{r} from different vantage points. These observations can be generated from a moving camera. Unfortunately, standard structure from motion (SFM) constraints based on epipolar geometry [13] cannot be used in this case: As the camera vantage point changes, reflected points move (flow) on the surface following the law of reflection; this violates the assumption of static 3D points which is essential in SFM methods. As shown in the pioneering studies by Koenderink and van Doorn [14], Blake [15], and Zisserman et al. [2], even when multiple observations of the same reflected point \mathbf{r} are available, \mathbf{r} and \mathbf{n}_r can still be estimated up to one-dimensional family of solutions

(whereas the local concave/convex shape ambiguity can be determined [15]). Similarly to the case of static cameras, additional constraints are required to estimate the unknown parameters. The theoretical analysis in Oren and Nayar [16] and more recently the level set formulation by Solem et al. [17] demonstrate that camera movements allow to recover 3D surface profiles if boundary conditions at object occluding contours are used. Bonfort and Sturm [18] develop a discrete multi-view approach where the surfaces reflect a calibrated 3D scene using a volumetric stereo framework [19]. Roth and Black [20] relax the assumption of calibrated scene by introducing a probabilistic formulation-based expectation-maximization that combines cues from specular and diffuse components, the former being defined as *specular flow*.

Application

Modeling the shape of reflective surfaces is valuable in numerous research and industrial applications such as digital archival (e.g., acquisition of digital models for preservation of artistic artifacts with reflective components), medicine (e.g., noninvasive inspection of organs such as the cornea of the eye [7]), and metrology of industrial parts.

References

1. Savarese S, Chen M, Perona P (2005) Local shape from mirror reflections. *Int J Comput Vision* 64(1):31–67
2. Zisserman A, Giblin P, Blake A (1989) The information available to a moving observer from specularities. *Image Video Comput* 7:38–42
3. Ikeuchi K (1981) Determining surface orientation of specular surfaces by using the photometric stereo method. *IEEE J Pattern Anal Mach Intell* 3:661–669
4. Torrance K, Sparrow E (1976) Theory for off-specular reflection from roughened surfaces. *J Opt Soc Am* 57: 1105–1114
5. Healey G, Binford T (1988) Local shape from specularity. *Comput Vision Graphics Image Process* 42:62–86
6. Nayar SK, Sanderson AC, Weiss LE, Simon DA (1990) Specular surface inspection using structured highlight and gaussian images. *T-RA* 6:208–218
7. Halstead M, Barsky B, Klein S, Mandell R (1996) Reconstructing curved surfaces from specular reflection patterns using spline surface fitting of normals. In: *ACM SIGGRAPH*, New Orleans, pp 335–342
8. Tarini M, Lensch H, Goesele M, Seidel H (2002) Shape from distortion: 3d range scanning of mirroring objects. In: *Proceedings of the SIGGRAPH*, sketches and applications. ACM, New York, p 248
9. Zheng J, Murata A (2000) Acquiring a complete 3d model from specular motion under the illumination of circular-shaped light sources. *IEEE J Pattern Anal Mach Intell* 8:913–920
10. Rozenfeld S, Shimshoni I, Lindenbaum M (2010) Dense mirroring surface recovery from 1d homographies and sparse correspondences. *IEEE Trans Pattern Anal Mach Intell* 99:325–337
11. Steger E, Kutulakos KN (2008) A theory of refractive and specular 3d shape by light-path triangulation. *Int J Comput Vision* 76(1):13–29
12. Adato Y, Vasilyev Y, Ben-Shahar O, Zickler T (2007) Towards a theory of shape from specular flow. In: *Proceedings of the IEEE international conference on computer vision (ICCV)*, Rio de Janeiro
13. Hartley RI, Zisserman A (2004) *Multiple view geometry in computer vision*, 2nd edn. Cambridge University Press, Cambridge, ISBN:0521540518
14. Koenderink J, van Doorn A (1980) Photometric invariants related to solid shape. *Optica Acta* 27:981–996
15. Blake A (1985) Specular stereo. In: *IJCAI*, Los Angeles, pp 973–976
16. Oren M, Nayar SK (1997) A theory of specular surface geometry. *Int J Comput Vision* 24(2):105–124
17. Solem J, Aanæs H, Heyden A (2003) A variational analysis of shape from specularities using sparse data. In: *International symposium on 3D data processing visualization and transmission*, Padova, Italy
18. Bonfort T, Sturm P (2003) Voxel carving for specular surfaces. In: *Proceedings of the IEEE international conference on computer vision (ICCV)*, Nice, pp 394–403
19. Kutulakos KN, Seitz SM (1999) A theory of shape by space carving. In: *Proceedings of the seventh IEEE international conference on computer vision*, Kerkyra, Greece, pp 307–313
20. Roth S, Black MJ (2006) Specular flow and the recovery of surface structure. In: *Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR)*, New York, pp 1869–1876

Shock Graph

Sven J. Dickinson¹, Ali Shokoufandeh² and Kaleem Siddiqi³

¹Department of Computer Science, University of Toronto, Toronto, ON, Canada

²Department of Computer Science, Drexel University Philadelphia, PA, USA

³School of Computer Science, McGill University, Montreal, PQ, Canada

Related Concepts

► [Many-to-Many Graph Matching](#); ► [Object Class Recognition \(Categorization\)](#)

Definition

The shock graph is obtained from the 2D Blum medial axis by incorporating properties of the radius function along the skeleton. The direction in which the radius function increases, or equivalently, the direction of the grassfire flow, is used to order groups of skeletal points and to derive parent-child relationships. This results in a directed acyclic graph whose nodes represent skeletal points and whose edges represent adjacency relationships. A variant of this construction associates skeletal points with edges, with the nodes representing the adjacencies.

Background

When Blum conceived of the medial axis or skeleton, his goal was to use it as a means to categorize objects from their projected (2D) outlines [4]. Specifically, by associating the direction of increasing radius value along a skeletal branch, or equivalently the direction of propagation of singularities of the grassfire flow, he proposed the concept of an axis-morphology or *a-morph* by which to achieve object categorization. His basic insight was that this could lead to a decomposition that reflected the qualitative part structure of the object. As an example, ignoring their detailed boundary geometry, outlines of hands would have similar *a-morphs* and these would be quite distinct from those of outlines of humans, fish or other object classes. In fact, he drew upon these later examples towards the end of his classic paper [4], where he also sketched possible extensions to 3D.

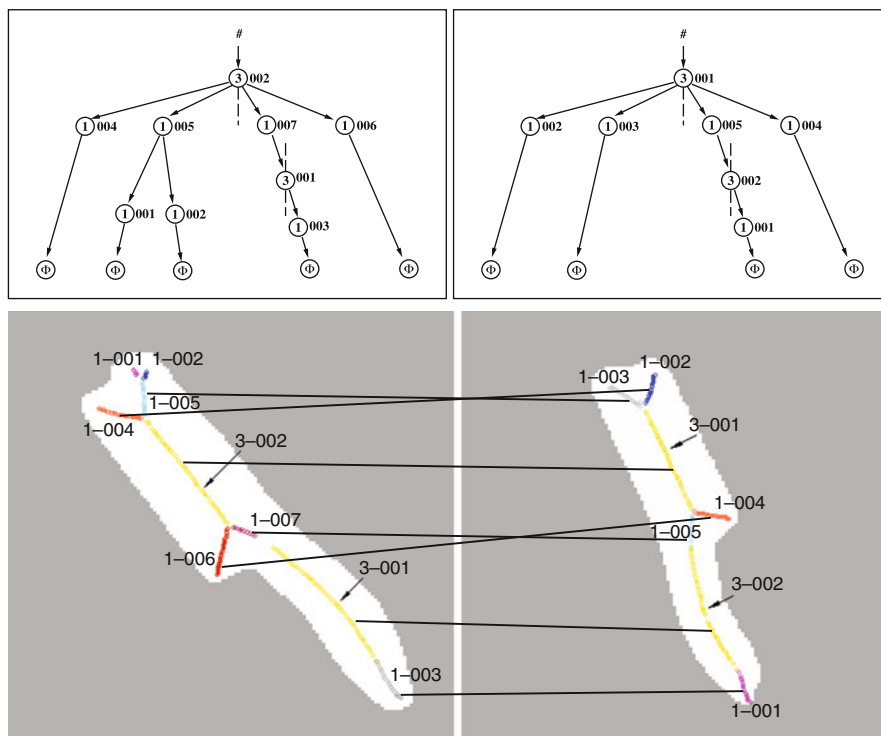
Whereas much has been written about medial or skeletal representations over the years (see [23] and also the medial axis/skeleton entry in this encyclopedia) the idea that an *a-morph* was essentially a directed graph which could be used for object recognition caught on only in the early 1990s. One likely reason is that it took the image analysis and computer vision communities many years to develop robust algorithms for skeleton computation. Since this time, however, a variety of successful approaches to view-based recognition using shock-graphs have been proposed and have been validated on large databases. Several of these are described in the present entry. There also

exist more recent variants of the shock graph, such as Macrini et al.'s bone graph [14], which attempt to mitigate the representational instability of the Blum medial axis. In fact, the mapping of the Blum skeleton to a graph-based representation, of which the shock graph is the most widely researched example, remains an active area of investigation.

Theory

The Blum medial axis or skeleton of a 2D outline is homotopic to it and is comprised of three types of skeletal points: endpoints of skeletal curves, interior points and branch points. The branch points are generically of degree 3, i.e., three skeletal curves are connected at a branch point. A formal classification is presented in [11]. The shock graph takes the 2D skeleton of a simple closed curve as input (one without holes) and labels each skeletal point according to whether the radius function at it is increasing monotonically (a 1-shock), is a local minimum (a 2-shock), is constant (a 3-shock) or is a local maximum (a 4-shock). Groups of adjacent 1-shocks are considered together, as are groups of 3-shocks. Given this interpretation, a directed acyclic graph is obtained by considering the skeletal points with the largest radii, which are the last to form in the grassfire flow, as the children of a dummy root node. The children are then placed, recursively, in order of decreasing radius value. This process of reversing the grassfire flow and adding 1-shock groups or 3-shock groups as children, is governed by the rules of a grammar, as shown in [24].

Rather than provide all the details of the grammar in this entry, the reader is referred to the examples in Fig. 1, which show the construction of the shock graphs of two brush shapes. The medial axis of each object is shown in the bottom row, with distinct groups of shocks being given a unique color (3-shocks are shown in yellow). In the labeling, the shock type appears first, followed by a unique identifier. The associated shock graphs are shown in the top row. It is clear that each shape is abstracted by a single root node (the 3-shock group describing the elongated portion of the brush), with its children being additional protrusions (1-shock groups). One of these protrusions has a 3-shock group as a child, which describes the handle of each brush. From this example it is evident



Shock Graph, Fig. 1 The shock graphs derived for two different views of a brush using the algorithm of Siddiqi et al. [24] are represented in the top row. The bottom row depicts the

correspondences between nodes in the shock graphs computed by the matching algorithm

that the shock graph is a formalization of Blum's a-morph, with the advantage that it lends itself to the use of graph-based methods for object categorization, as detailed below.

It is also important to point out that there is a variant of the shock graph where the representation places the skeletal points at edges of the graph, with the nodes representing connections. This variant is described in detail in [18, 19]. This representation has lead to different but equally successful methods for object recognition, based on a notion of the edit-distance between two graphs. The results of using this approach are also briefly described below.

Shock Graph-Based Object Categorization

An object categorization system based on shock graphs consists of two components: (1) an indexing component, which takes an input shock graph and returns, from a large database of model shock graphs, a small

number of candidate shock graphs that might account for the input; and (2) a matching component, which takes one of the candidates and the input, and computes a similarity (or distance), along with a set of node correspondences. Under ideal conditions, the input shock graph would contain no artifacts due to noise, occlusion, or clutter, and would be isomorphic to one of the model shock graphs (provided that the input object represents one of the model objects). However, such conditions are highly unlikely, for in addition to noise, occlusion, and scene clutter, ligature-induced instabilities [1] often lead to spurious nodes/edges as well as medial branch oversegmentation. Formulating the problem as graph isomorphism, subgraph isomorphism, or even largest isomorphic subgraph will not lead to a meaningful solution, for large, or even significant isomorphisms may simply not exist between two shock graphs that represent instances of the same category. The shock graph indexing and matching problems are therefore inexact graph indexing/matching problems.

Indexing Shock Graphs

Given an input shock graph, the goal of the indexing module is to quickly (sublinearly) retrieve a small number of candidate model database shock graphs among which the input is likely included. As mentioned above, the input shock graph may be corrupted in a number of ways, precluding a simple global (based on the entire input) indexing framework. For example: (1) occlusion may remove part of the input shock graph and replace the missing part with a shock graph (or subgraph) belonging to a different object; (2) shadows or poor illumination may simply delete some portion of the input shock graph; (3) scene clutter may embed the object shock graph (or portion thereof) in a much larger “scene” shock graph; and (4) ligature-based instability may introduce spurious nodes or may overpartition other nodes in the input shock graph. These factors require a part-based indexing framework that can operate in the presence of noise, occlusion, clutter, and ligature-based instability.

One such indexing framework that is applicable to not only shock graphs but any hierarchical, graph-based representation (specifically, any directed acyclic graph-based representation) was introduced by Shokoufandeh et al. [22], originally for the purpose of shock graph indexing. The key concept behind the approach is to capture the abstract shape of a graph (or subgraph) with a low-dimensional vector, yielding an efficient indexing mechanism. Capturing the abstract shape of a graph is important so that the index is invariant to noise and minor within-class shape deformation. Indexing at the part level is important in the presence of occlusion and scene clutter. Mapping a discrete graph structure to a low-dimensional point facilitates a simple nearest-neighbor search in a geometric space for similar model parts which, in turn, can vote for those model objects that contain those parts. Those model objects receiving the largest votes represent those candidate objects passed to the shock graph matching module for a more detailed analysis.

The graph-based shape abstraction is computed at every non-leaf node, and captures the abstract “shape” of the underlying subgraph rooted at that node. Therefore, each non-leaf node (with only four shock graph node types, leaf nodes are far too uninformative/ambiguous) “votes” for those objects that share its substructure; the root of the graph would therefore vote at the object level, and would be

meaningful only if the object were unoccluded and not embedded in a larger scene. Mapping the structure of a rooted subgraph to a vector assigned to the subgraph’s root is based on a spectral analysis of the graph’s structure. The eigenvalues of a graph’s adjacency matrix (whose values are 0,1,−1) capture important properties of the degree distribution of the graph’s nodes. The eigenvalues can be combined to yield a low-dimensional abstraction of the graph’s shape in terms of how and where the edges are distributed throughout the graph. Moreover, such a spectral “signature,” called the *topological signature vector*, is proven to be stable under minor perturbations of graph structure due to noise. Details of the approach are found in [22], while an application of the same indexing framework to a different hierarchical graph, specifically a 3-D medial surface graph, can be found in [25].

Matching Shock Graphs

Given two shock graphs, e.g., one representing the input and one representing a model candidate, the matching component needs to return not only a similarity or distance measure that can be used to rank order the candidates, but also an explicit correspondence that defines which model nodes correspond to which input nodes. Such correspondence is necessary, for in the case of a cluttered scene, those nodes found to match a given model would be removed, and another candidate model matched to the remaining nodes. Moreover, the correspondence need not be one-to-one, for in the case of ligature-induced medial branch oversegmentation, node correspondence may be *many-to-many*.

Siddiqi et al. [24] developed a matching algorithm for shock graphs which, like the indexing framework of Shokoufandeh et al. [22] discussed above, can be applied to the matching of any directed acyclic graph structure, provided that a domain-dependent node similarity function is given. The algorithm is based on the same spectral graph theoretic abstraction that forms the heart of the indexing component described above. The algorithm formulates the matching of two graphs as finding a maximal matching in a bipartite graph over the two nodes sets (input and model). The edge weights (each spanning one input shock graph node and one model shock graph node) in the graph have two components: (1) the distance between the two nodes’ respective topological similarity vectors, defining the similarity of their underlying graph structures

(rooted at the two nodes); and (2) a node similarity function (the only domain-dependent component of the algorithm) that defines the similarity of the node attributes (for shock graphs, this encodes the geometric similarity between the two skeletal branches corresponding to the two nodes).

At first glance, the matching algorithm would seem to throw out all the important hierarchical structure in the two graphs (absent in the bipartite graph); nodes in one graph are matched to nodes in the other graph, but the edges in the two original graphs appear to play no role. However, the key contribution of the algorithm is that the hierarchical edge structure is brought back via the topological signature vector similarity term. For the bipartite matching algorithm to match two nodes (i.e., select that edge in the matching), both their geometric similarity and their topological similarity must be high. In other words, the contents of the two nodes must be similar and the subgraphs rooted at the two nodes must be similar. The algorithm iterates by computing a matching, selecting the best edge from the matching (having maximum similarity), adding it to the solution set, and recursively continuing the process on the remaining graphs (after removing the pair of matching nodes defined by the best edge). Details of the approach are found in [24], while its application to other shape matching problems is described in [21] (multiscale blob and ridge graphs), [25] (medial surface graphs), and [8, 26] (curve skeleton graphs).

























The above algorithm eventually yields a one-to-one node correspondence between the two graphs. However, because the algorithm generates the node correspondence in a coarse-to-fine manner, stopping the algorithm at the level of a coarse node-to-node correspondence defines an explicit many-to-many correspondence between the nodes in the subgraphs rooted at the coarse nodes. Moreover, since the topological signature vectors are stable under small amounts of additive graph noise, similarity can remain high even though the two subgraphs may have different numbers of nodes. As the cardinalities of the two graphs' node sets begin to differ more dramatically, for example due to heavy under- or over-segmentation, the method breaks down and more powerful many-to-many graph matching must be employed.

One such method for many-to-many graph matching of medial axis-based graphs was proposed by Demirci et al. [9, 10]. Their algorithm transforms the

graphs into a finite dimension metric space in which an approximate solution to the many-to-many matching problem becomes tractable. The embedding step will result in a set of points, each representing a vertex of the original graph. Their proposed embedding has the additional property that pairwise distances between points in the target metric space closely resemble the shortest-path distances between the corresponding nodes in the graphs. Matching two graphs can then be formulated as the problem of matching their two embeddings. The many-to-many matching of the two embeddings then can be computed by solving a transportation problem using the Earth Mover's Distance algorithm [7]. The solution of this latter problem computes the mass which flows from one weighted point set to another that minimize the total transportation cost. The computed flows, in turn, define the many-to-many node correspondences between the original graphs.

The problem of matching shock graphs has also been studied in the context of edit-distance methods [18, 29]. These algorithms estimate the cost of matching as a function of edit operations, including node relabelings, additions and deletions, and edge contraction that transform one graph into another. A fundamental issue in devising algorithms based on edit-distance is the choice of cost of each operation. Torsello and Hancock [29] use the heuristic proposed by Bunke [5] for the cost associated with their edit operations. For example, the cost of relabeling elements is less than the cost of performing a deletion followed by inserting a new node with a new label. In contrast, Sebastian et al. [18] propose a multi-step heuristic to derive their edit costs. Their overall heuristic is centered around the notion of a shape cell, i.e., a collection of shapes which have identical shock graph topology. They define the cost of the deformation operation as a function of the discrepancy between matching shock attributes of shapes within a given cell. The cost associated with other edit operations is derived as the limit of the deformation cost when a shape moves to the boundary a shape cell.

Caelli and Kosinov [6] show how inexact matching can be utilized for measuring shape similarity between shock graphs. Their method establishes correspondence between sets (clusters) of vertices of two given graphs and as such can be viewed as a many-to-many matching approach. Their algorithm can be viewed as

Instance	Distance to Class Prototype								
									
	0.02	2.17	4.48	3.55	2.96	0.21	4.58	14.33	10.01
	2.39	0.10	5.97	15.90	3.98	0.14	26.12	17.28	28.94
	10.89	4.72	2.08	12.24	3.12	2.15	19.73	10.11	12.64
	7.15	6.42	1.19	1.35	5.10	3.38	10.58	11.11	11.11
	4.08	7.72	2.98	1.49	4.26	4.14	26.60	13.54	14.21
	14.77	6.72	5.69	0.36	2.30	5.90	10.58	16.25	19.10
	7.86	8.90	5.94	0.74	1.59	1.10	10.81	10.39	16.08
	2.66	4.23	3.23	6.47	0.62	1.48	11.73	15.38	15.15
	3.18	5.31	1.25	4.64	0.60	1.30	14.18	17.22	9.08
	4.55	0.76	1.32	2.86	1.49	0.11	21.38	15.35	13.04
	6.77	19.46	22.11	13.27	8.21	29.50	0.15	5.12	5.03
	8.73	23.14	31.45	24.41	10.16	31.08	0.18	8.45	7.05
	12.46	19.0	27.40	14.58	24.26	17.10	8.85	7.49	16.93
	13.86	23.07	12.81	11.24	17.48	23.23	6.02	6.92	3.06
	15.73	21.28	14.10	12.46	19.56	19.21	9.53	7.12	5.06

Shock Graph, Fig. 2 Similarity between database and class prototypes computed using the algorithm of Siddiqi et al. [24]. In each row, a box is drawn around the most similar shape

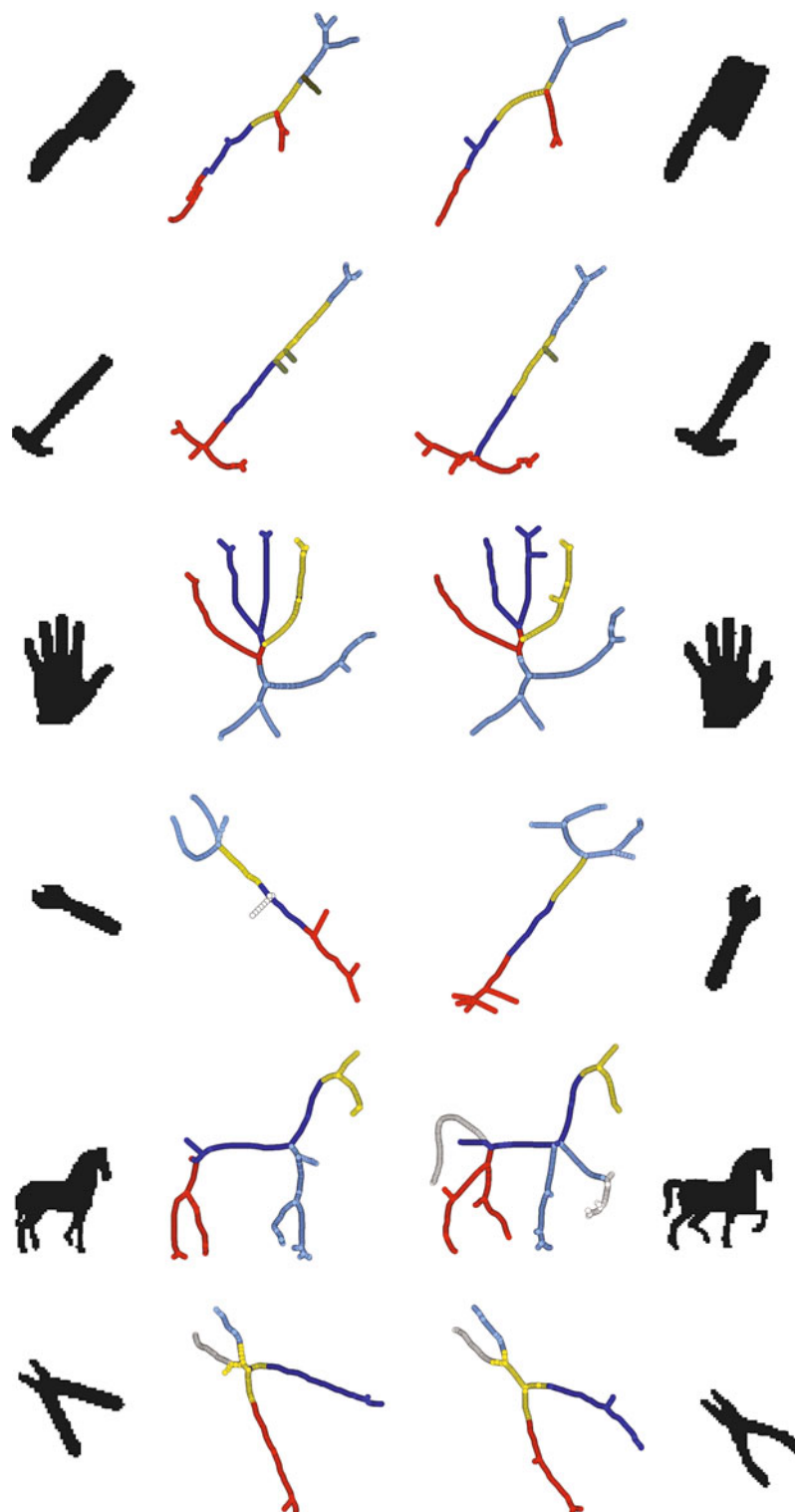
a generalization of the approach of Scott and Longuet-Higgins [17]. The actual matching is established using the renormalization of projections of vertices into the eigenspaces of graphs combined with a form of relational clustering. Similar to other inexact matching algorithms, their eigenspace renormalization projection clustering method is able to match graphs with different numbers of vertices.

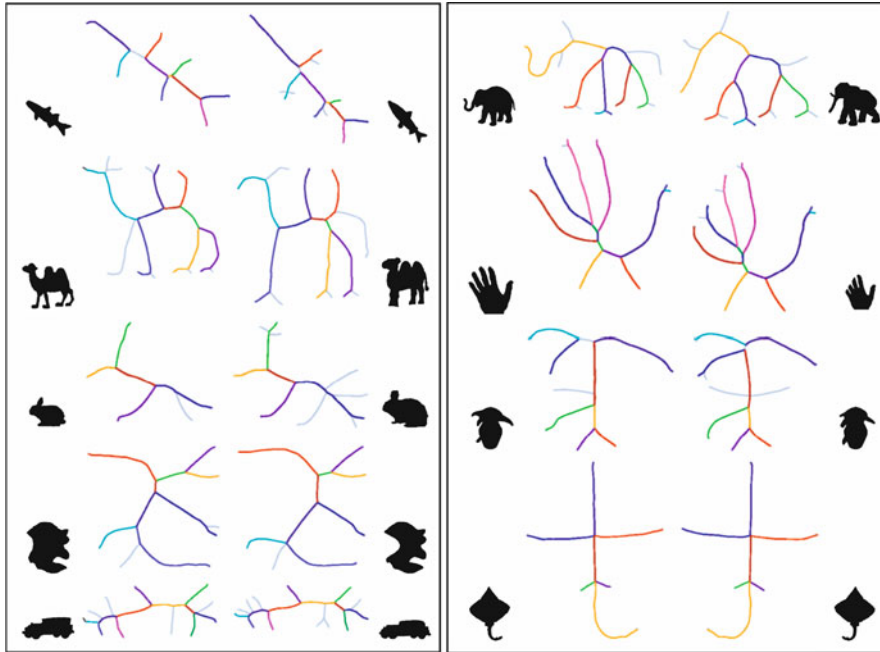
Experimental Results

This section presents some examples of shock graphs and their matchings using the approaches described above. Figure 1(top) illustrates two shock graphs, describing different views of a brush, computed by the algorithm of Siddiqi et al. [24]. The underlying shocks, along with the computed matchings between segments (nodes), are shown in Fig. 1(bottom).

Figure 2 represents the ability of the algorithm to compare objects based on their prototypical or coarse shape. Here, columns 2 through 10 denote the prototype views for each of nine object classes. The similarity between the prototypes and some of the objects in the database is reflected in the rows of this table. For each row, a box has been placed around the most similar shape. Demirci et al. [9] also evaluated the effectiveness of their matching algorithm for shape retrieval based on shock graphs from the Rutgers Tool Database [24]. Figure 3 shows some examples of the many-to-many feature matching results obtained from the algorithm for some of the objects in the Rutgers Tools Database. Finally, Fig. 4 shows the results obtained from applying the edit-distance algorithm of Sebastian et al. [18] to the matching of shock segments. Note that their edit distance algorithm will also produce a sequence of intermediate shock graphs that identify the steps of the transformation of one input shock graph to another.

Shock Graph, Fig. 3 The results of matching skeleton graphs for some pairs of shapes in the Rutgers Tools Database using the algorithm of Demirci et al. [9]. Corresponding segments are shown using the same color. Observe that correspondences are intuitive in all cases





Shock Graph, Fig. 4 The matching results for a few shock graphs produced by the edit-distance algorithm of Sebastian et al. [18]. Matching shock branches are shown using the same

color, while the *gray* colored edges in the shock graphs indicate that they are spliced or contracted

Open Problems

Symmetry is a powerful shape regularity that has formed the basis of many shape representations, including generalized cylinders [3], superquadrics [16], and geons [2]. Just as geons provide a qualitative and discrete shape abstraction of a generalized cylinder, shock graphs provide a discrete and qualitative shape abstraction of a medial axis. The resulting graph is ideally suited to shape categorization, for it is part-based, is stable under within-class deformation, and is stable under part articulation. However, the shock graph also faces some important challenges. First of all, it assumes that a closed contour has been recovered from an image, separating figure from background. While figure-ground segmentation remains an open research problem, it is important to note that in a categorization system, a perfect figure-ground separation may not be necessary. If a significant portion of the figure's boundary is correctly segmented, a significant portion of the resulting shock graph may be correct – enough to yield the correct candidate (among the list of returned candidates) during indexing. Still, while a shock graph does preserve

locality of representation, significant figure-ground segmentation errors can propagate through the representation, disrupting it to a degree that prevents effective indexing. A recent attempt to recover a symmetric part decomposition from a cluttered scene has been reported by Levinshtein et al. [13], in which symmetric parts are detected locally (bottom-up) and then grouped to form an approximation to a medial axis.

The second challenge facing the shock graph is the ligature-based instability discussed earlier [1]. A number of approaches exist to try and regularize the medial axis through boundary smoothing, e.g., [12, 20, 27]; however, these methods do not effectively address the ligature structure. Other methods have sought to abstract the medial axis by regularizing out small internal branches, e.g., [28, 30]; however these methods don't explicitly target ligature structure. A recent promising approach to abstracting out ligature structure is proposed by Macrini et al. [14, 15], yielding a representation, called the *bone graph*, whose parts are the non-ligature medial branches that represent the salient parts and whose edges represent the “glue” (defined by the ligature branches) that binds the parts.

References

- August J, Siddiqi K, Zucker S (1999) Ligature instabilities in the perceptual organization of shape. *CVIU* 76(3): 231–243
- Biederman I (1985) Human image understanding: Recent research and a theory. *Comput Vis Graph Image Process* 32:29–73
- Binford TO (1971) Visual perception by computer. In: *Proceedings of the IEEE conference on systems and control*, Miami
- Harry Blum (1973) Biological shape and visual science. *J Theor Biol* 38:205–287
- Bunke H (1997) On a relation between graph edit distance and maximum common subgraph. *Pattern Recognit Lett* 18(8):689–694
- Caelli T, Kosinov S (2004) An eigenspace projection clustering method for inexact graph matching. *IEEE Trans Pattern Anal Mach Intell* 26(4):515–519
- Cohen SD, Guibas LJ (1999) The earth mover's distance under transformation sets. In: *Proceedings of the international conference on computer vision (ICCV)*, Kerkyra, pp 1076–1083
- Cornea N, Demirci MF, Silver D, Shokoufandeh A, Dickinson S, Kantor P (2005) 3d object retrieval using many-to-many matching of curve skeletons. In: *Proceedings of the international conference on shape modeling and applications (SMI)*, MIT, Cambridge, pp 368–373
- Demirci MF, Shokoufandeh A, Keselman Y, Bretzner L, Dickinson S (2006) Object recognition as many-to-many feature matching. *Int J Comput Vis* 69(2):203–222
- Demirci F, Shokoufandeh A, Dickinson S (2009) Skeletal shape abstraction from examples. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31(5): 944–952
- Giblin Peter J, Kimia Peter J (2003) On the local form and transitions of symmetry sets, medial axes, and shocks. *IJCV* 54(1–3):143–157
- Katz Robert A, Pizer Stephen M (2003) Untangling the blum medial axis transform. *Int J Comput Vis* 55(2–3): 139–153 .
- Levinstein A, Sminchisescu C, Dickinson S (2009) Multi-scale symmetric part detection and grouping. In: *Proceedings of the international conference on computer vision (ICCV)*, Kyoto
- Macrini D, Dickinson S, Fleet D, Siddiqi K (2011) Bone graphs: Medial shape parsing and abstraction. *Comput Vis Image Underst (CVIU)*, special issue on graph-based representations, Special Issue on Graph-Based Representations, 115(7):1044–1061
- Macrini D, Dickinson S, Fleet D, Siddiqi K (2011) Object categorization using bone graphs. *Computer Vision and Image Understanding (CVIU)* 115(8):1187–1206
- Pentland A (1986) Perceptual organization and the representation of natural form. *Artif Intell* 28:293–331
- Scott G, Longuet-Higgins H (1991) An algorithm for associating the features of two patterns. *Proceedings of royal society of london B* 244:21–26
- Sebastian T, Klein P, Kimia B (2001) Recognition of shapes by editing shock graphs. In: *Proceedings of the international conference on computer vision (ICCV)*, Vancouver 755–762
- Sebastian T, Klein P, Kimia B (2004) Recognition of shapes by editing shock graphs. *IEEE Trans Pattern Anal Mach Intell* 26:550–571
- Doron Shaked, Bruckstein Alfred M (1998) Pruning medial axes. *Comput Vis Image Underst* 69(2):156–169
- Shokoufandeh A, Bretzner L, Macrini D, Demirci MF, Jönsson C, Dickinson S (2006) The representation and matching of categorical shape. *Comput Vis Image Underst* 103(2):139–154
- Shokoufandeh A, Macrini D, Dickinson S, Siddiqi K, Zucker SW (2005) Indexing hierarchical structures using graph spectra. *IEEE Trans Pattern Anal Mach Intell* 27(7):1125–1140
- Siddiqi K, Pizer Stephen M (2008) *Medial representations: mathematics, algorithms and applications*. Springer, Dordrecht
- Siddiqi K, Shokoufandeh A, Dickinson S, Zucker S (1999) Shock graphs and shape matching. *Int J Comput Vis* 30: 1–24
- Siddiqi K, Zhang J, Macrini D, Shokoufandeh A, Bioux S, Dickinson S (2008) Retrieving articulated 3-d models using medial surfaces. *Mach Vis Appl* 19(4):261–275
- Sundar H, Silver D, Gagvani N, Dickinson S (2003) Skeleton based shape matching and retrieval. In: *Proceedings of the international conference on shape modelling and applications (SMI)*, Seoul, pp 130–142
- Hüseyin Tek, Kimia Benjamin B (2001) Boundary smoothing via symmetry transforms. *J Math Imaging Vis* 14(3):211–223
- Telea A, Sminchisescu C, Dickinson S (2004) Optimal inference for hierarchical skeleton abstraction. In: *Proceedings of the international conference on pattern recognition*, Cambridge, UK, pp 19–22
- Torsello A, Hancock ER (2003) Computing approximate tree edit distance using relaxation labeling. *Pattern Recognit Lett* 24(8):1089–1097
- van Eede M, Macrini D, Telea A, Sminchisescu C, Dickinson S (2006) Canonical skeletons for shape matching. In: *Proceedings of the international conference on pattern recognition*, Hong Kong, pp 64–69

Shot Noise

► [Photon, Poisson Noise](#)

Simulated Annealing

Juergen Gall

Max Planck Institute for Intelligent Systems,
Tübingen, Germany

Synonyms

[Monte Carlo annealing](#); [Probabilistic hill climbing](#);
[Statistical cooling](#); [Stochastic relaxation](#)

Definition

Simulated annealing is a stochastic computational technique derived from statistical mechanics for finding near globally-minimum-cost solutions to large optimization problems [1].

Background

Many computer vision problems require the minimization of an application dependent objective function in a high-dimensional state space subject to conflicting constraints. Finding the global minimum can be an NP-complete problem since the objective function tends to have many local minima. A procedure for solving hard optimization problems should sample values of the objective function in such a way as to have a high probability of finding a near-optimal solution and should also lend itself to efficient implementation. A method which meets these criteria was introduced by Kirkpatrick et al. [2] and independently by Černý [3] in the early 1980s. They introduced the concepts of annealing in combinatorial optimization. These concepts are based on a strong analogy between the physical annealing process of solids and the problem of solving large combinatorial optimization problems.

Theory

Statistical mechanics is the study of the behavior of very large systems of interacting components, such as atoms in a fluid, in thermal equilibrium at a finite temperature. If the system is in thermal equilibrium at a given temperature T , then the probability $\pi_T(s)$ that the system is in a given state s depends upon the energy $E(s)$ of the state and follows the Boltzmann distribution:

$$\pi_T(s) = \frac{\exp\left(-\frac{E(s)}{k_B T}\right)}{\sum_{s' \in \Omega} \exp\left(-\frac{E(s')}{k_B T}\right)}, \quad (1)$$

where k_B denotes a physical constant known as the Boltzmann constant and Ω the set of all possible states.

Using a technique developed by Metropolis et al. [4], one can simulate the behavior of a system

of particles in thermal equilibrium at temperature T . Suppose that at time t the system is in state q_t . Then, a subsequent state r_{t+1} is generated by a perturbation mechanism which transforms the current state into a next state. If the energy difference, $E(r_{t+1}) - E(q_t)$, is less than or equal 0, the state r_{t+1} is accepted as current state. Otherwise, the state r_{t+1} is accepted with probability

$$p = \frac{\pi_T(r_{t+1})}{\pi_T(q_t)} = \exp\left(-\frac{E(r_{t+1}) - E(q_t)}{k_B T}\right). \quad (2)$$

Since the method accepts states that decrease the energy as well as those that increase the energy, it is the principle that avoids entrapment at a local minimum. It can be shown that as $t \rightarrow \infty$, the probability that the system is in a given state s equals $\pi_T(s)$, and thus that the distribution of states generated converges to the Boltzmann distribution [5].

In order to obtain a low-energy state of the energy function E , one must use an annealing process, where the temperature of the system is elevated, and then gradually lowered, spending enough time at each temperature to reach thermal equilibrium. Applying simulated annealing to an optimization problem in computer vision where the energy function becomes the objective function to minimize, requires two ingredients:

- An annealing schedule consisting of a starting temperature, a decreasing set of temperatures, and the amount of time to spend at each temperature;
- A perturbation mechanism that generates new states.

The annealing algorithm proposed by Kirkpatrick et al. [2] consists of running the Metropolis-Hastings algorithm [4, 6] at each temperature in the annealing schedule for the amount of time prescribed by the schedule, and selecting the final state as a near-optimal solution.

Geman and Geman [5] applied simulated annealing to image restoration and determined an annealing schedule sufficient for convergence. Specifically, for a given sequence of temperatures $\{T_t\}$ such that $T_t \rightarrow 0$ as $t \rightarrow \infty$ and $T_t \geq \frac{c}{\log(t)}$ for a large constant c , the probability that the system is in configuration s as $t \rightarrow \infty$ is equal to $\pi_0(s)$. For a finite set Ω , the optimal annealing schedule for the convergence of the generated states to the set of global minima with probability 1 was determined by Hajek [7].

Simulated annealing is not limited to discrete state spaces. It can also be applied to minimize objective functions defined on Euclidean spaces, i.e., $\Omega \subseteq \mathbb{R}^d$, where similar convergence results have been proved [8]. Further convergence results and detailed discussions on simulated annealing are given in the books [9, 10].

In the discrete case, the perturbation mechanism depends usually on the application where some examples are given in [9]. In the simplest case, a new state is randomly sampled from a local neighborhood of the previous state, e.g., by permutations, swapping, or inversions. In the continuous case, the visiting distribution can be modeled as Gaussian distribution that favors local search, the algorithm is also called Boltzmann machine, but also other distributions like the Cauchy-Lorentz distribution, known as fast simulated annealing [11], can be applied. This distribution results in frequently local searches, but can also generate a state that is very distant to the current state. Having the state q_t , a new state $r_{t+1} = q_t + \Delta s$ is generated by sampling from

$$g_t(\Delta s) = \frac{1}{(\pi T_t)^{\frac{d}{2}}} \exp\left(-\frac{\|\Delta s\|^2}{T_t}\right) \quad (\text{Boltzmann machine}), \quad (3)$$

$$g_t(\Delta s) = \frac{\Gamma(\frac{d+1}{2})}{\pi^{\frac{d+1}{2}}} \frac{T_t}{(\|\Delta s\|^2 + T_t^2)^{\frac{d+1}{2}}} \quad (\text{Cauchy machine}). \quad (4)$$

The new state is then accepted according to the probability p (2). An approach that covers Gaussian and Cauchy-Lorentz distribution as special cases is called generalized simulated annealing [12]. The generalized acceptance probability (2) reads

$$p = \left(1 + \frac{(a-1)(E(r_{t+1}) - E(q_t))}{T_t^a}\right)^{-\frac{1}{a-1}}, \quad (5)$$

the generalized visiting distribution is defined by

$$g_t^v(\Delta s) = \left(\frac{v-1}{\pi}\right)^{\frac{d}{2}} \frac{\Gamma(\frac{1}{v-1} + \frac{d-1}{2})}{\Gamma(\frac{1}{v-1} - \frac{1}{2})} \frac{(T_t^v)^{-\frac{d}{3-v}}}{\left(1 + (v-1) \frac{\|\Delta s\|^2}{(T_t^v)^{\frac{2}{3-v}}}\right)^{\frac{1}{v-1} + \frac{d-1}{2}}}, \quad (6)$$

and the generalized annealing schedule is given by

$$T_t^b = T_1^b \frac{2^{b-1} - 1}{(1+t)^{b-1} - 1} \quad b \in \{a, v\}. \quad (7)$$

The additional parameters (a, v) give an additional flexibility where $(1, 1)$ corresponds to the Boltzmann machine and $(1, 2)$ to the Cauchy machine. In practice, the optimal choice of (a, v) depends on the objective function and needs to be empirically determined.

For speeding up simulated annealing, the algorithm can be implemented in parallel [9]. The annealing principle is also used for other optimization methods. For instance, a deterministic annealing method has been proposed for clustering [13] or an interacting particle system with annealing properties has been proposed in [14] and applied to human motion capture [15], where at each iteration a set of particles estimates the current distribution.

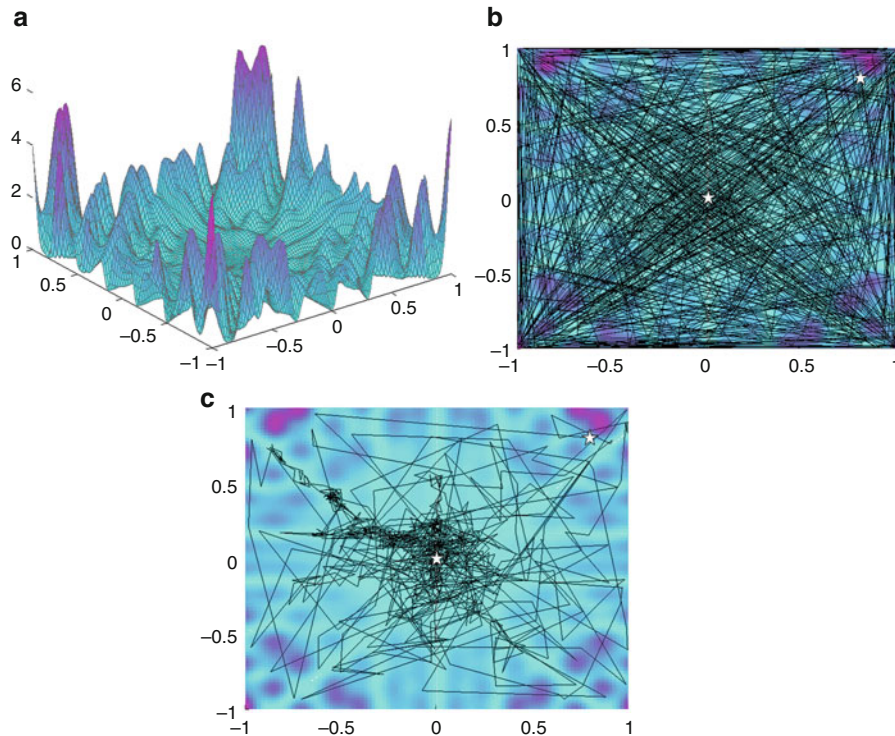
Application

Since its introduction in 1983, simulated annealing has been applied for solving complex, non-convex optimization problems in image processing and computer vision. An overview of applications in the 1980s including the work of Geman and Geman [5] is given in [9]. Nowadays, simulated annealing is still an easy-to-implement and practically useful tool for solving a wide spectrum of optimization problems, particularly for solving hard problems where no completely successful heuristics exist. For an optimal performance, however, the algorithm needs to be tailored to the problem at hand.

Experimental Results

The following example for global optimization in \mathbb{R}^2 is taken from [16]: Let

$$h(x, y) = (x \sin(20y) + y \sin(20x))^2 \cdot \cosh((\sin(10x)x + (x \cos(10y) - y \sin(10x))^2 \cdot \cosh(\cos(20y)y)) \quad (8)$$



Simulated Annealing, Fig. 1 (a) Function $h(x, y)$. (b) Visited points using Boltzmann machine and initial temperature 100. The starting point, (0.8, 0.8), and the final estimate are indicated

by a *white star*. (c) Visited points using Cauchy machine. The Cauchy machine cools down faster, focusing the search more on low energy regions

Simulated Annealing, Table 1 Average and standard deviation for the obtained estimate (\hat{x}, \hat{y}) , its function value $h(\hat{x}, \hat{y})$, and the number of required function evaluations at different initial temperatures T_1 .

	T_1	(\hat{x}, \hat{y})	$h(\hat{x}, \hat{y}) \times 10^5$	# function evaluations
Boltzmann	1	$(-0.02, 0.04) \pm (0.23, 0.37)$	1.14 ± 2.00	$1,681.9 \pm 507.5$
Cauchy	1	$(0.42, 0.38) \pm (0.38, 0.43)$	700.61 ± 3241.27	$1,504.4 \pm 461.0$
Boltzmann	10	$(-0.04, -0.01) \pm (0.38, 0.51)$	7.16 ± 12.01	$1,740.9 \pm 573.0$
Cauchy	10	$(0.00, -0.02) \pm (0.46, 0.39)$	0.20 ± 0.55	$1,732.0 \pm 580.7$
Boltzmann	100	$(-0.02, 0.03) \pm (0.34, 0.33)$	2.41 ± 4.02	$1,733.9 \pm 560.8$
Cauchy	100	$(0.00, 0.02) \pm (0.19, 0.24)$	0.50 ± 1.12	$1,802.6 \pm 494.0$

be the objective function to minimize (Fig. 1). Its global minimum is 0, attained at $(x, y) = (0, 0)$.

In Table 1, results are shown for the Boltzmann and the Cauchy machine. The starting point was (0.8, 0.8) and each simulation was repeated 100 times. Each simulation was stopped when the average change in value of the objective function in 1,000 iterations was less than 10^{-6} . The table contains the average and standard deviation for the obtained estimate (\hat{x}, \hat{y}) , its function value $h(\hat{x}, \hat{y})$, and the number of necessary function evaluations.

References

1. Davis L (1987) Genetic algorithms and simulated annealing. Morgan Kaufmann, San Francisco
2. Kirkpatrick S, Jr CG, Vecchi M (1983) Optimization by simulated annealing. Science 220(4598):671–680
3. Černý V (1985) Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. J Optim Theory Appl 45(1):41–51
4. Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E (1953) Equations of state calculations by fast computing machines. J Chem Phys 21(6): 1087–1092

5. Geman S, Geman D (1984) Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans Pattern Anal Mach Intell* 6:721–741
6. Hastings W (1970) Monte carlo sampling methods using markov chains and their applications. *Biometrika* 57(1): 97–109
7. Hajek B (1988) Cooling schedules for optimal annealing. *Math Oper Res* 13(2):311–329
8. Royer G (1989) A remark on simulated annealing of diffusion processes. *SIAM J Control Optim* 27(6):1403–1408
9. Aarts E, Korst J (1989) Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing. Wiley, New York
10. Gidas B (1995) 7: Metropolis-type Monte Carlo simulation algorithms and simulated annealing. In: *Topics in contemporary probability and its applications*. CRC, Boca Raton, pp 159–232
11. Szu H, Hartley R (1987) Fast simulated annealing. *Phys Lett A* 122:157–162
12. Tsallis C, Stariolo D (1996) Generalized simulated annealing. *Physica A* 233:395–406
13. Rose K (1998) Deterministic annealing for clustering, compression, classification, regression, and related optimization problems. *Proc IEEE* 86(11): 2210–2239
14. Moral PD (2004) Feynman-Kac formulae. Genealogical and interacting particle systems with applications. Springer, New York
15. Gall J, Rosenhahn B, Seidel HP (2008) An Introduction to Interacting Simulated Annealing. In: *Human motion – understanding, modeling, capture and animation*. Springer, Dordrecht, pp 319–343
16. Robert C, Casella G (2002) Monte Carlo statistical methods. Springer, New York

Single Viewpoint

► [Center of Projection](#)

Situation Graph Trees

Christian Micheloni¹ and Gian Luca Foresti²

¹Department of Mathematics and Computer Science, University of Udine, Udine, Italy

²Department of Mathematics and Computer Science, University of Udine, Udine, Italy

Synonyms

[Generically describable situation](#); [Situation scheme](#)

Definition

Situation Graph Trees (SGT) provide a deterministic formalism to represent the knowledge required for human behavior modeling.

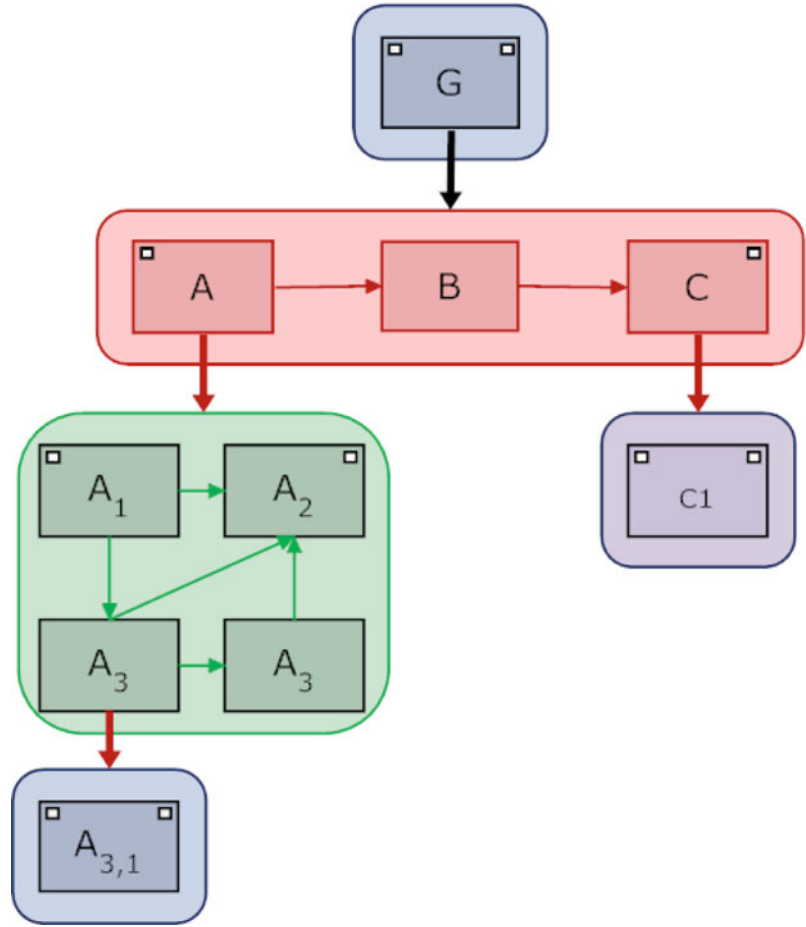
Background

In many domains, high-level descriptions to represent the status of an environment are desirable. Describing a situation requires to conceptualize the knowledge about the possible actions of the actors involved in the environment and their possible interactions. Conceptual descriptions of different application domains like traffic analysis [1], parking lot security [2, 3], and human behavior recognition [4] are of primary importance. The conceptualization can proceed from simple descriptions (simple events) to complex descriptions (complex events). Following relations, concepts can be aggregated into more complex concepts. Hence, an event can be described as a sequence of simple events. To allow such an incremental description of the events, two main processes are required: (a) modeling of the behaviors and (b) the reasoning engine. Different approaches like Bayesian networks [5], Hidden Markov models [6], and SVM [7] have been recently introduced. However, in complex environments, such models are difficult to apply as they would require a large amount of data for training the models. In such cases, a formalism to model behaviors by means of temporal and semantic relationships or specialization of fundamental concepts can be useful. Situational Graph Trees (SGTs) represent such a modeling tool.

Theory

The Situation Graph Trees (SGTs) [4] provide a deterministic formalism to represent the knowledge necessary to describe an actor behavior. Generally, SGTs are based on the description of the situation that consists on an agent state and the possible actions that the agent can actuate in such a state. Thus, a hierarchy of possible situation is defined on temporal and conceptual terms. This means that given a recognized situation only, its possible successor is evaluated at the next time instant.

**Situation Graph
Trees, Fig. 1** Example of
SGT



The fundamental block of the SGT is the situation scheme that represents the knowledge of an agent for a given time instant. A situation scheme is composed by sections:

- State: describe the state of an agent in terms of predicates
- Action: describe the possible and supposed actions that an agent can do whenever one of the state predicates is satisfied

Situation schemes are connected by means of direct edges (called *prediction edges*) to define a temporal successor relationship between situation schemes. When an agent is instantiated by its predicates, a possible next situation is represented by a scheme pointed by a prediction edge originated by the current situation. If an agent keeps on staying in its current situation for more than a single time instant, *self-prediction edges*, starting from a situation and pointing to itself, are used to model such a behavior. Thus, a situation scheme can be a ring of a chain of situation schemes describing a sequence of situations. Such sequences,

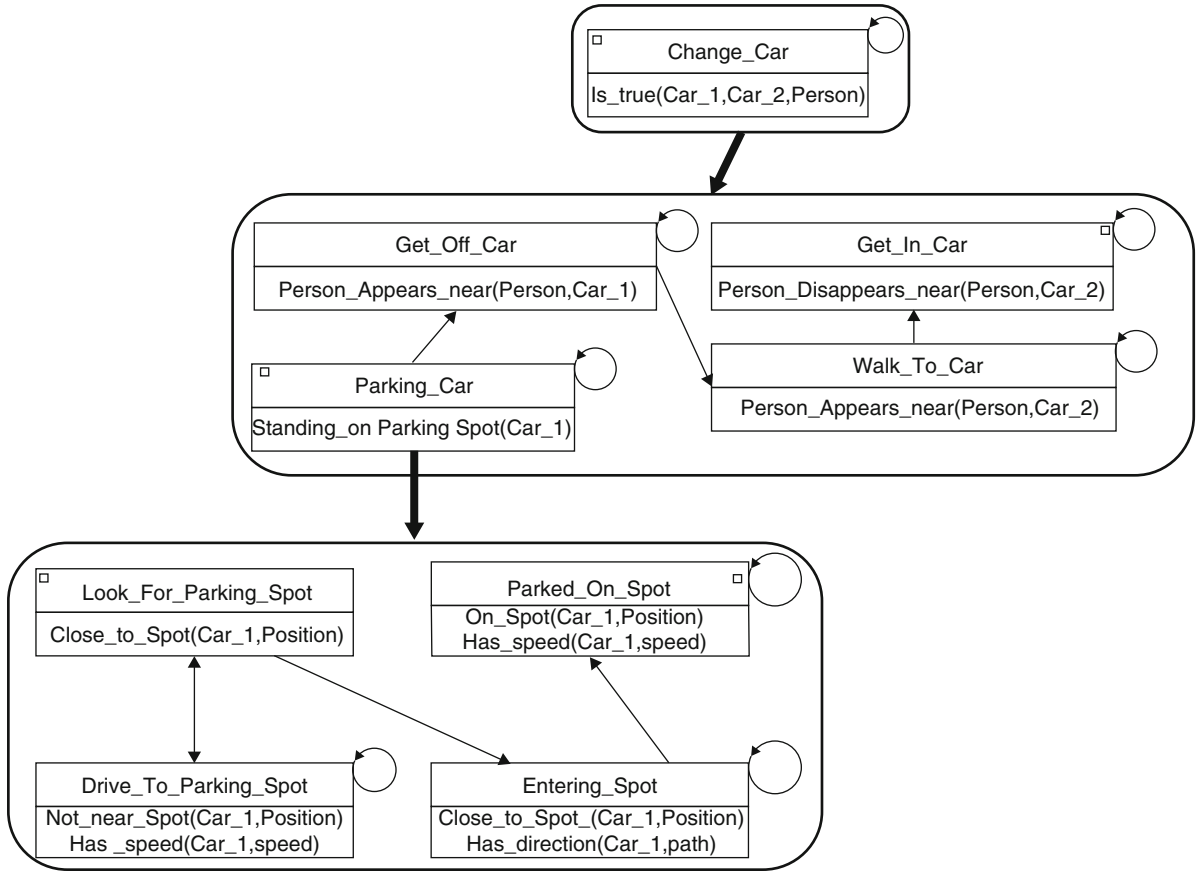
composed by situation schemes and prediction edges, are called situation graphs. A situation can be temporally or conceptually refined by particularizing its situation scheme. In such a case, a situational graph is connected to a situational scheme by means of a particularized edge. Following [4], it is possible to derive the following definitions for SGTs.

Definition 1 (SGT-Episode) *Any sequence E of situations inside a situation graph G that is a path from a start situation to a end situation is defined as SGT-Episode.*

Definition 2 (Particularized SGT-Episode) *Any SGT-Episode of a situation graph G particularizing a situation scheme s is defined as a particularizing SGT-Episode.*

Definition 3 (SGT-Event) *Given a SGT T , an event is defined as:*

- Any SGT-Episode E within the root situation graph of T .
- Given an SGT-Event E , replacing any situation S in E with a particularized SGT-Episode of S is a new SGT-Event.



Situation Graph Trees, Fig. 2 Example of SGT for an event of interest in a parking lot monitoring context

Definition 4 (Particularized SGT-Event) *Given a SGT T and a SGT-Event E , a SGT-Event \hat{E} is defined as a particularized SGT-Event of E iff \hat{E} is obtained from E by substituting any situation scheme S of E with a particularized SGT-Episode of S .*

Definition 5 (Maximal Event) *Given a SGT T , any SGT-Event E is a maximal event iff there not exists a situation scheme S in E that can be substituted with a particularized SGT-Episode of S .*

Definition 6 (Compatible Events) *Given a SGT T , E and E' are compatible events iff:*

- Each situation scheme in E is a situation scheme in E' .
- For each pair (S_i, S_j) in E where S_i proceed S_j , there is the same pair in E' .

Representation

The common representation of SGTs is given on Fig. 1. Situation schemes are represented as rectangles and situation graphs as set of rectangles with rounded corners. Start situations (end situations) are depicted with small rectangles in the upper left (right)

corner of the situation schemes. Prediction edges are thin arrows that decide the current situation and the possible next situation. Thick arrows represent particularization edges.

Application

A common application of SGTs is the representation of behaviors in video-surveillance context. In such a field, contrarily to anomaly detection algorithm, expected behaviors or behaviors of interest can be defined for all the actors/agents operating inside the monitored environment. To represent such behaviors, SGT can be powerfully exploited. As example in a parking lot, the monitoring application may be interested in detecting and recognizing a person driving into a parking spot, leaving the parked car, and driving away on board of a second car. Such an event can be described by the SGT depicted in Fig. 2 that can be further particularized in the situation schemes of the first child node.

References

1. Haag M, Nagel H (2000) Incremental recognition of traffic situations from video image sequences. *Image Vis Comput* 18(2):137–153
2. Micheloni C, Remagnino P, Eng HL, Geng J (2010) Intelligent monitoring of complex environments. *Intelligent Systems*, IEEE 25(3):12–14
3. Micheloni C, Snidaro L, Foresti G (2009) Exploiting temporal statistics for events analysis and understanding. *Image Vis Comput* 27(10):1459–1469
4. Arens M, Nagel H (2003) Behavioral knowledge representation for the understanding and creation of video sequences. In: German conference on artificial intelligence, Hamburge(GE), pp 149–163
5. Robertson N, Reid I (2006) A general method for human activity recognition in video. *Comput Vis Image Underst* 104(2–3):232–248
6. Galata A, Johnson N, Hogg D (2001) Learning variable-length markov models of behavior. *Comput Vis Image Underst* 81(3):398–413
7. Piciarelli C, Micheloni C, Foresti G (2008) Trajectory-based anomalous event detection. *IEEE Trans Circuits Syst Video Technol* 18(11):1544–1554

Situation Scheme

► Situation Graph Trees

Solid Texture

► Bidirectional Texture Function and 3D Texture

Space Curves

► Curves in Euclidean Three-Space

Sparse Coding

John Wright
Visual Computing Group, Microsoft Research Asia,
Beijing, China

Synonyms

[Sparse representation](#)

Definition

Sparse coding is the act of expressing a given input signal (e.g., image or image patch) as a linear superposition of a small set of basis signals chosen from a prespecified dictionary.

Background

At a high level, the problem of sparse coding is one of representing a given input signal as efficiently as possible:

Given an input signal $\mathbf{y} \in \mathbb{R}^m$ (say an image or image patch) and a dictionary of basis signals $\mathbf{a}_1 \dots \mathbf{a}_n \in \mathbb{R}^m$, find a good approximation

$$\mathbf{y} \approx x_1 \mathbf{a}_1 + x_2 \mathbf{a}_2 + \dots + x_n \mathbf{a}_n$$

in which most of the coefficients x_i are zero.

That is, we try to represent \mathbf{y} as a linear combination of basis elements in which only a few of the coefficients are nonzero (i.e., the vector $\mathbf{x} = (x_1 \dots x_n)$ is *sparse*). This deceptively simple problem arises repeatedly in signal processing, modern statistics, and machine learning. The most readily apparent application is in data compression, where we can consider the sparse coefficients \mathbf{x} as a compressed representation of the signal \mathbf{y} . However, numerous additional applications arise in signal and image acquisition, denoising, and inpainting. Sparse coding techniques have also received significant attention in the statistics literature, where sparsity is recognized as a means of regularizing high-dimensional inference – in particular, for regularizing linear regression when the number of predictors is larger than the number of observations.

These interactions make sparse coding a very vibrant area of research, with contributions from statistics, signal processing, optimization, applied mathematics, and cognitive neuroscience. Indeed, the term “sparse coding” originally comes from the neuroscience literature, where it has been observed that seeking a sparse codes for natural image patches yields Gabor-like basis functions that resemble the receptive fields in the human visual system [1]. In applied mathematics and statistics, a deep literature has developed around the question of when it is possible to solve sparse coding problems efficiently.

While the tools and problems encountered in sparse coding have precedents dating back almost a century, much of the development has been relatively recent. In computer vision, techniques from sparse coding (and related areas of sparse error correction and compressed sensing) have been employed for recognizing faces and objects, performing image upsampling, denoising, and superresolution. At the time of writing this article, sparse coding techniques are the subject of intense exploration in the vision community [2].

Theory

The model problem in sparse coding is one of searching for the *sparsest representation* of a given input signal \mathbf{y} as a linear combination of dictionary elements:

$$\text{minimize } \|\mathbf{x}\|_0 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y}. \quad (1)$$

Above, the ℓ^0 -pseudonorm

$$\|\mathbf{x}\|_0 = \#\{i \mid x_i \neq 0\}$$

simply counts the number of elements in \mathbf{x} that are not zero. Although conceptually desirable, the problem (1) is computationally intractable (hard to approximate in the worst case), and so it is common practice to replace the ℓ^0 norm with a more tractable surrogate. One way to do this is to instead minimize the ℓ^1 -norm

$$\|\mathbf{x}\|_1 = \sum_i |x_i|.$$

This gives a convex optimization problem

$$\text{minimize } \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{y}. \quad (2)$$

This relaxation is well motivated, in the sense that the ℓ^1 norm can be shown to be the tightest convex underestimator of the ℓ^0 norm over the set of all vectors \mathbf{x} with $\max_i |x_i| \leq 1$. Moreover, whereas the non-convex cardinality minimization problem (1) does not admit an efficient algorithm, the convex problem (2) can be cast as a linear program and solved efficiently. For more details on available techniques for solving

the optimization problem (2), the interested reader can refer to the survey paper [3].

In certain situations, the link between (2) and (1) can be made quite a bit tighter. When the dictionary \mathbf{A} satisfies technical conditions that essentially assert that its columns are not too collinear, it can be shown that these two problems are formally equivalent: the tractable optimization (2) exactly recovers the sparse solution to (1)! For example, suppose that the columns \mathbf{a}_i of \mathbf{A} have unit ℓ^2 norm, and let

$$\mu(\mathbf{A}) \doteq \max_{i \neq j} |\langle \mathbf{a}_i, \mathbf{a}_j \rangle|. \quad (3)$$

Then, whenever $\mathbf{y} = \mathbf{A}\mathbf{x}_0$ for some \mathbf{x}_0 satisfying

$$\|\mathbf{x}_0\|_0 < \frac{1}{2}(1 + 1/\mu(\mathbf{A})), \quad (4)$$

we have that \mathbf{x}_0 is the unique optimal solution to the ℓ^1 minimization (2). That is to say, whenever there exists a sufficiently sparse solution to the system of equations $\mathbf{y} = \mathbf{A}\mathbf{x}$, this solution will be recovered by ℓ^1 norm minimization. Variants of this result have been obtained by a number of authors; the version described above is due to Donoho and Elad [4]. There is a vast literature on guarantees for ℓ^1 -minimization – in particular, a family of beautiful results on ℓ^1 -minimization with random matrices \mathbf{A} has inspired the recent development of *compressed sensing*, an approach to more efficiently acquire signals that are sparse in some known basis. For readers who are interested in learning more, one starting point is the survey paper [5].

In practice, the observation \mathbf{y} may contain noise, and so it is desirable to relax the constraint $\mathbf{A}\mathbf{x} = \mathbf{y}$. This leads to a new convex program

$$\text{minimize } \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2 \leq \varepsilon, \quad (5)$$

which is known in the signal processing literature as *basis pursuit denoising* [6]. Its Lagrangian reformulation,

$$\text{minimize } \|\mathbf{x}\|_1 + \lambda \|\mathbf{A}\mathbf{x} - \mathbf{y}\|_2^2/2, \quad (6)$$

is known as the *Lasso* in statistics [7]. The optimization problems (5) and (6) are equivalent under an appropriate calibration $\varepsilon \leftrightarrow \lambda$, although no explicit expressions for the corresponding parameters are known.

The theoretical results alluded to above make ℓ^1 -minimization a very attractive approach to sparse

coding. However, it is by no means the only available algorithm. Researchers have explored a number of nonconvex objective functions which may more closely approximate the ℓ^0 norm in (1), at the expense of being hard to solve in general. Examples include the ℓ^p norms $\|\mathbf{x}\|_p = (\sum_i x_i^p)^{1/p}$ ($0 < p < 1$) and entropy-like functions such as $\sum_i \log(1 + |x_i|^2)$.

Greedy algorithms comprise another popular alternative to ℓ^1 -minimization. These algorithms construct a solution \mathbf{x} via an iterative procedure that repeatedly selects a “best” dictionary element \mathbf{a}_i to add to the representation. One prototypical example is the Orthogonal Matching Pursuit (OMP) algorithm [8] (which has been rediscovered in a number of different settings). OMP maintains an active set of indices J and a residual \mathbf{r} . Initially, J is empty, and $\mathbf{r} = \mathbf{y}$. At each step, the index $j \in \{1 \dots n\}$ that maximizes $|\langle \mathbf{a}_j, \mathbf{r} \rangle|$ is added to the active set J . The sparse coefficients \mathbf{x} are estimated via

$$\text{minimize } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \mathbf{x}(J^c) = \mathbf{0}. \quad (7)$$

The residual \mathbf{r} is updated as $\mathbf{r} = \mathbf{y} - \mathbf{A}\mathbf{x}$. This procedure is repeated until a sufficiently accurate approximation to \mathbf{y} is obtained. OMP is attractive for its simplicity and also comes with performance guarantees in some situations; the interested reader can refer to [5] and the references therein for a start.

All of the algorithms described above assume that the given observation \mathbf{y} indeed has a sparse approximation in terms of a known dictionary \mathbf{A} . In some situations, this can be guaranteed from prior understanding of the physical structure of the problem. For example, in face recognition, the dictionary \mathbf{A} can be constructed from images of training faces chosen in order to guarantee a good approximation of the test image \mathbf{y} [9]. In other situations, it may be possible to design optimal representations for classes of signals – as witnessed by the development of signal representations in signal processing and harmonic analysis over the past few decades.

However, if the signal model is not known ahead of time, or if the specific class of signals is believed to have some additional structure, an attractive alternative is to attempt to learn the dictionary \mathbf{A} itself from sample data. This leads to a problem known as *dictionary learning*, in which we observe multiple examples $\mathbf{Y} = \mathbf{y}_1 \dots \mathbf{y}_p \in \mathbb{R}^m$. The goal is to find a dictionary

$\mathbf{A} \in \mathbb{R}^{m \times n}$ of basis functions and sparse coefficient vectors $\mathbf{X} = \mathbf{x}_1 \dots \mathbf{x}_p \in \mathbb{R}^n$ such that $\mathbf{Y} \approx \mathbf{A}\mathbf{X}$:

$$\text{minimize}_{\mathbf{A}, \mathbf{X}} \|\mathbf{X}\|_0 \quad \text{subject to} \quad \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F \leq \varepsilon. \quad (8)$$

Notice that here the minimization is with respect to both \mathbf{A} and \mathbf{X} , and so even if we relax the ℓ^0 norm, the resulting optimization problem,

$$\text{minimize}_{\mathbf{A}, \mathbf{X}} \|\mathbf{X}\|_1 + \lambda \|\mathbf{Y} - \mathbf{A}\mathbf{X}\|_F^2 / 2, \quad (9)$$

is not convex, and guaranteeing a global optimum is difficult. The key observation is that if either \mathbf{A} or \mathbf{X} is fixed, the optimization (9) becomes convex in the remaining variable. This naturally suggests an alternating directions approach:

$$\mathbf{X}_{k+1} = \arg \min_{\mathbf{X}} \|\mathbf{X}\|_1 + \lambda \|\mathbf{Y} - \mathbf{A}_k \mathbf{X}\|_F^2 / 2 \quad (10)$$

$$\mathbf{A}_{k+1} = \arg \min_{\mathbf{A}} \|\mathbf{X}_{k+1}\|_1 + \lambda \|\mathbf{Y} - \mathbf{A} \mathbf{X}_{k+1}\|_F^2 / 2. \quad (11)$$

Each of these subproblems can be solved efficiently. It is not difficult to show that this procedure converges to some pair $(\mathbf{A}_*, \mathbf{X}_*)$. However, unlike the problem of sparse coding in a known dictionary \mathbf{A} (as discussed above), for dictionary learning there is currently no theory to explain when the algorithm will succeed. This is partially a consequence of the fact that the unknowns \mathbf{A} and \mathbf{X} enter into the equation $\mathbf{Y} = \mathbf{A}\mathbf{X}$ in a bilinear fashion – the dictionary learning problem is difficult to analyze for the same reason that it is difficult to solve.

Nevertheless, empirical evidence suggests that there are situations in which this approach learns very effective data representations, and many researchers have used dictionary learning techniques to solve problems in image processing and vision. In fact, much of the initial excitement about sparsity in vision came from the classical paper of Olshausen and Field [1], which observed that the dictionary elements \mathbf{a}_i learned from natural images patches are similar to the receptive fields in the human visual system.

Many variants of the basic alternating directions approach have been investigated in the literature. For a more thorough history and additional references, we refer the interested reader to the survey paper [10].

Application

As alluded to above, at the time of this writing, numerous applications of sparse coding are being explored in the computer vision community. Due to their sheer number, most of these works lie beyond the scope of this article. In this section we briefly outline two examples of how the algorithms for sparse coding can be useful for solving problems in imaging and vision. The interested reader is invited to see [2] for a more thorough review.

Our first example comes from automatic face recognition [9]. In this application, the “dictionary elements” \mathbf{a}_i are training images of subjects in the database. Several images of each subject are taken under varying illumination, stacked as vectors in \mathbb{R}^m (here $m = W \times H$ is the number of image pixels), concatenated together to form a large matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$. Given a new test image \mathbf{y} taken under new illumination, and possibly subject to some additional corruption or occlusion, one can solve a sparse coding problem:

$$\text{minimize } \|\mathbf{x}\|_1 + \|\mathbf{e}\|_1 \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{e}. \quad (12)$$

Here, the sparse “error” term \mathbf{e} allows some robustness to occlusion, while the sparse coefficients \mathbf{x} naturally select images of the same subject to participate in the representation of \mathbf{y} . This sparse code can be used for identifying the subject pictured in \mathbf{y} or rejecting impostors not present in the database; see [9] for details.

Another representative example comes in image inpainting and superresolution [11, 12]. Suppose that patches \mathbf{y} of a given input image are known to have good sparse approximations in some dictionary \mathbf{A} , learned from a large collection of natural image patches. Suppose that some of the image pixels missing, so that rather than observing $\mathbf{y} \in \mathbb{R}^m$, we observe only a subset $\mathbf{y}(\Omega)$, where $\Omega \subset \{1 \dots m\}$. Then, one very natural approach to recovering the missing pixels is to solve a sparse coding problem (In [11], greedy algorithms are used, rather than ℓ^1 -minimization.)

$$\text{minimize } \|\mathbf{x}\|_1 \quad \text{subject to } \|\mathbf{y}(\Omega) - \mathbf{P}_\Omega \mathbf{A}\mathbf{x}\|_2^2 \leq \varepsilon^2, \quad (13)$$

where $\mathbf{P}_\Omega \in \mathbb{R}^{|\Omega| \times m}$ is a projection matrix onto the coordinates indexed by Ω . Once the solution $\hat{\mathbf{x}}$ is recovered, one can estimate the missing elements via $\hat{\mathbf{y}} = \mathbf{A}\hat{\mathbf{x}}$. For more examples of how sparse coding algorithms can be used in inpainting and related problems, see [11].

Open Problems

At the time of writing this article, there are numerous open problems in sparse coding, many of which are currently under vigorous attack. One question that has received significant recent attention is how to incorporate additional structure or prior knowledge into the algorithm to allow more accurate recovery of the sparse coefficients \mathbf{x} . This leads to notions such as “group sparse coding,” in which certain subsets of coefficients are known to all be either active or inactive, simultaneously [13].

For some vision applications, explaining the good performance of sparse coding techniques and understanding their limitations remain open problems. At the same time, with so much recent development, we arguably have yet to fully realize the full power of sparsity for vision problems.

References

1. Olshausen B, Field D (1997) Sparse coding with an over-complete basis set: a strategy employed by V1? *Vis Res* 37(23):3311–3325
2. Wright J, Yang A, Mairal J, Sapiro G, Huang T, Yan S (2010) Sparse representation for computer vision and pattern recognition. *Proc IEEE* 98(6):1031–1044
3. Tropp J, Wright S (2010) Computational methods for sparse solution of linear inverse problems. *Proc IEEE* 98(6): 948–958
4. Donoho D, Elad M (2003) Optimally sparse representation in general (non-orthogonal) dictionaries via ℓ^1 -minimization. *Proc Natl Acad Sci* 100:2197–2202
5. Bruckstein AM, Donoho DL, Elad M (2009) From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Rev* 51(1):34–81
6. Chen S, Donoho D, Saunders M (2001) Atomic decomposition by basis pursuit. *SIAM Rev* 43(1):129–159
7. Tibshirani R (1996) Regression shrinkage and selection via the Lasso. *J R Stat Soc B* 58(1):267–288
8. Pati Y, Rezaifar R, Krishnaprasad P (1993) Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In: *Proceedings of*

- the Asilomar conference on signals, systems and computers, Pacific Grove, CA. vol 1, pp 40–44
9. Wright J, Yang A, Ganesh A, Sastry S, Ma Y (2009) Robust face recognition via sparse representation. *IEEE Trans Pattern Anal Mach Intell* 31(2):210–227
 10. Rubenstein R, Bruckstein A, Elad M (2010) Dictionaries for sparse representation modeling. *Proc IEEE* 98(6): 1045–1057
 11. Mairal J, Elad M, Sapiro G (2008) Sparse representations for color image restoration. *IEEE Trans Image Process* 17(1):53–69
 12. Yang J, Wright J, Huang T, Ma Y (2010) Image super-resolution via sparse representation. *IEEE Trans Image Process* 19(11):2861–2873
 13. Yuan M, Lin Y (2007) Model selection and estimation in regression with grouped variables. *J R Stat Soc B* 68(1): 49–67

Sparse Representation

► Sparse Coding

Spatiotemporal Reasoning

David Young
School of Informatics, University of Sussex, Falmer,
Brighton, UK

Definition

Spatiotemporal reasoning is the use of time-varying information in image sequences, together with assumptions about the properties of objects and the environment, to make predictive inferences or to test hypotheses about the scene.

Background

Spatiotemporal reasoning plays an important rôle in theories of human and animal vision, especially those associated with the ecological psychology school of J.J. Gibson and his followers [6]. A central idea is that the optic flow field provides active organisms with the means to pick up the *affordances* of their environment (the likely possibilities for future interaction with the environment). A seminal example is the case of an object approaching to the viewer along the line

of sight and without rotation: in this case, the rate of expansion of the object's image specifies the inverse of the expected time to collision, assuming that the relative velocity remains constant.

The essential point of this example is that the time to collision may be estimated from the changing image without having any information about positions or velocities and without estimating these variables. A spatiotemporal relationship in the image (rate of expansion) is linked to a spatiotemporal property of the scene (time to collision) with no need for any geometrical reconstruction. This is particularly relevant to systems where rapid control of action in response to visual information is required, and the relationship has been proposed as part of the mechanism controlling interceptive actions in people and animals, as well as for the automatic control of vehicles [1, 7].

In computer vision, the expansion/time-to-collision relationship has been explored theoretically [8], while practical methods for measuring expansion rates have been investigated [2, 9, 11]. However, this particular relationship can best be seen as a special case of the general theory describing the optic flow induced by the relative motion of a camera and a surface, laid out by Koenderink and coworkers [4, 5]. This theory allows predictive inferences to be made without assuming line-of-sight approach or pure translation.

In parallel with such quantitative reasoning, qualitative spatiotemporal reasoning has also been investigated. Here, object attributes such as identity are assigned partly on the basis of expectations generated from the history of the image sequence. An example is the logging of vehicle movements at an airport, exploiting expectations about the spatiotemporal structure of event sequences [3, 10].

Although qualitative and quantitative spatiotemporal reasoning have very different flavors, both aim to extract practically useful information from image sequences by exploiting such properties of the scene as continuity of motion, the solidity and compactness of objects, and surface smoothness. The area is thus closely related to object tracking and robot navigation.

Theory

First-order optic flow provides an example of the use of spatiotemporal information. It is straightforward to

establish the relationship between the parameters of first-order flow models and information that may be applied to the control of action.

We assume that the optic flow in a region of an image may be approximated by a first-order model; thus,

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} D + S_1 & S_2 - R \\ S_2 + R & D - S_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

where $\begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$ is the optic flow vector at image position $\begin{bmatrix} x_1 & x_2 \end{bmatrix}^T$ and $\begin{bmatrix} v_1 & v_2 \end{bmatrix}^T$ is the optic flow at the origin of image coordinates (the “zero-order” flow). D , R , S_1 , and S_2 are the parameters of the first-order component of the flow: respectively, dilation, rotation, and the two components of shear.

If we also assume that the flow is generated by a smooth surface patch in the scene, we can define some parameters which provide a partial description of the current state of relative motion between the patch and the camera. For example, the plane *immediacy* is the reciprocal of the time remaining until the camera passes through the plane tangent to the surface patch, assuming constant velocity. For a parachutist looking at a patch anywhere on a planar ground surface, it is the reciprocal of the expected time to landing. It is possible to show (using, e.g., Koenderink’s framework [4]), that plane immediacy is computable from the optic flow parameters plus the component of the camera’s angular rotation rate about the line of sight (also called the *spin*). The equation is

$$i_P = D - 3S \cos \phi$$

where $S = \sqrt{S_1^2 + S_2^2}$, $\sin \phi = (R - \omega)/S$, and ω is the spin. The relationship for direct approach, $i_P = D$, is a special case, applicable when the shear component of the flow vanishes.

Similarly, the immediacy of the plane passing through the surface patch, and normal to the line of sight, may be determined from

$$i_A = D - S \cos \phi$$

The essential quality of these quantities is that no information about the spatial layout of the scene is

required or inferred. Nonetheless, they can be applied to the practical control of movement, such as braking a vehicle to a halt at an obstacle. In general, optic flow measurements constrain the physical motion parameters to some subspace without determining those parameters exactly; this constraint may nonetheless be sufficient information to solve practical navigational or interceptive tasks.

References

1. Alenya G, Nègre A, Crowley JL (2009) Time to contact for obstacle avoidance. In: European conference on mobile robotics, Dubrovnik, Croatia, 09 2009
2. Blake A, Curwen R, Zisserman A (1993) A framework for spatiotemporal control in the tracking of visual contours. *Int J Comput Vis* 11(2):127–145
3. Buxton H, Walker N (1988) Query based visual analysis: spatio-temporal reasoning in computer vision. *Image Vis Comput* 6(4):247–254
4. Koenderink JJ (1986) Optic flow. *Vis Res* 26(1):161–179
5. Koenderink JJ, van Doorn AJ (1992) Second-order optic flow. *J Opt Soc Am A* 9(4):530–538
6. Lee DN (1980) The optic flow field: the foundation of vision. *Philos Trans R Soc Lond Ser B* 290(1038):169–179
7. Lee DN, Reddish PE (1981) Plummeting gannets: a paradigm of ecological optics. *Nature* 293:293–294
8. Maybank S (1987) Apparent area of a rigid moving body. *Image Vis Comput* 5(2):111–113
9. Seales W (1995) Measuring time-to-contact using active camera control. In: Hlavác V, Šára R (eds) *Computer analysis of images and patterns. Lecture Notes in Computer Science*, vol 970. Springer, Berlin/Heidelberg, pp 944–949
10. Toal A, Buxton H (1992) Spatio-temporal reasoning within a traffic surveillance system. In: Sandini G (ed) *Computer Vision (ECCV’92). Lecture notes in computer science*, vol 588. Springer, Berlin/Heidelberg, pp 884–892
11. Tunley H, Young D (1994) First order optic flow from log-polar sampled images. In: *ECCV ’94: Proceedings of the 3rd European conference on computer vision*, Secaucus, NJ, USA, vol 1. Springer, New York, pp 132–137

SPDE

► [Stochastic Partial Differential Equations](#)

Specular Highlight

► [Specularity, Specular Reflectance](#)

Specularity, Specular Reflectance

Robby T. Tan

Department of Information and Computing Sciences,
Utrecht University, Utrecht, CH, The Netherlands

Synonyms

Interface reflection; MirrorLike reflection; Specular highlight

Related Concepts

►Dichromatic Reflection Model; ►Fresnel Equations;
►Surface Roughness

Definition

Specular reflection occurs when light is incident on a boundary interface between two different media and immediately reflects back to the medium where it comes from. Specular reflectance is the ratio of the reflected light by a boundary interface to the incident light. The visual appearance of specular reflections is known as specularity or specular highlight. To find the regions of surfaces that exhibit specular reflections is generally known as specularity detection.

Background

Reflection of light from an object is principally caused by the surface and body of the object. The former is known as specular or interface reflection and the latter is body or diffuse reflection. The specular reflection is the shiny mirrorlike reflection, which is commonly present in both man-made and natural objects. Mirrors, glass, ceramics, gold, silver, many fruits' skins, some leaves, etc. emit specular reflections. Theoretically, by considering the definition, almost all objects emit specular reflections, although the amount of the reflections varies depending on the object's optical properties, such as the surface roughness and the Fresnel reflection coefficient.

There are two main reasons why specular reflection is important in computer vision: (1) Many algorithms in computer vision assume perfect diffuse reflections and deem specular reflections to be outliers. However, in the real world, the presence of specular reflections is inevitable. Hence, incorporating the knowledge of specular reflections is essential to make the algorithms more robust. (2) Many computer vision algorithms may benefit from the information conveyed by specular reflection. This information includes the photometric and the geometric information, where the photometric information may be important for material recognition, and the geometric information may be useful for shape recognition.

Theory

Fresnel equations describe the reflection and transmission of light or electromagnetic waves at an interface between two media of differing refractive indices. According to the equations, when unpolarizing light impinges on a point of a flat and smooth surface, the reflection coefficient is $R = 0.5(R_{\perp} + R_{\parallel})$, where R_{\perp} and R_{\parallel} are the reflections of the electric field when it is perpendicular and parallel to the surface, respectively. The reflection coefficient is dependent on the angle of incoming light with respect to the surface normal and the refractive indices of the two media.

When a surface is perfectly flat and smooth (i.e., a perfect mirror), the direction of specular reflection will follow the law of reflection, which states the angle of incoming light θ_i and the angle of outgoing reflected light θ_r are the same ($\theta_i = \theta_r$). This implies that the specularly reflected light cannot be seen by an observer if the observer's position is not at the same direction as θ_r , which is true for the case of perfect mirrors. However, in many objects which are not perfect mirrors, a certain degree of specularity is still observable, even though the observer's position is slightly apart from the direction of θ_r . In other words, specular reflections do not only form a sharp line (spike) distribution of reflection but also form a lobe distribution. Therefore, there are two components of specular reflections: (1) specular spike and (2) specular lobe.

The Torrance-Sparrow reflection model [1] provides a good approximation of the specular lobe component, which is expressed as:



Specularity, Specular Reflectance, Fig. 1 (a) A dichromatic object exhibiting both diffuse and specular reflection. (b) The separated diffuse reflection component. (c) The separated specular reflection component. The results were computed using [5]

$$\rho = \frac{FG}{\cos \theta_r} \exp \left(-\frac{\alpha^2}{2\sigma^2} \right) \quad (1)$$

where F is the Fresnel reflection coefficient, G is the geometrical attenuation factor, θ_r is the angle between the viewing direction and the surface normal, α is the angle between the surface normal and the bisector of the viewing direction and the light source direction, and σ represents the surface roughness.

The Torrance-Sparrow reflection model uses geometric optics to describe the mechanism of specular reflection, which is only valid when the wavelength of light is much smaller than the roughness of the surface. According to [2], the model uses a slope distribution model to represent the profile of a surface. The surface is assumed to be a collection of planar microfacets, where their dimension is much larger than the wavelength of incident light. Each microfacet is perfectly smooth, and the orientation of each facet deviates from the mean orientation of the surface by an angle α . The model considers the masking and shadowing of microfacets by adjacent facets, where they can block light going into a facet or light reflected by it. The geometrical attenuation factor, G , is introduced to compensate the masking and shadowing effect. The surface roughness, σ , represents the spatial distribution of the lobe. The larger the value of σ , the larger the lobe distribution (implying less shiny surfaces), and vice versa. In this reflection model, the distribution of the specular reflections follows the Gaussian distribution, with mean α and standard deviation σ . Later, Cook and Torrance [3] replaced the Gaussian distribution with the Beckmann distribution function.

While the Torrance-Sparrow reflection model is able to approximately generate a mirrorlike distribution, namely, when σ is considerably small, its main drawback is that it cannot generate both the specular spike and specular lobe at the same time. To overcome this drawback, Nayar et al. [2] introduced a model unifying the Torrance-Sparrow's specular lobe and a spike specular model, and the latter is modeled as:

$$K_{ss} \delta(\theta_i - \theta_r) \delta(\phi_r) \quad (2)$$

where K_{ss} is the strength of the specular spike and (θ_r, ϕ_r) is the direction of the reflected. Nayar et al. base their analysis on the Beckmann-Spizzichino reflection model [4], which predicts the presence of both the specular lobe and spike. Unlike the Torrance-Sparrow reflection, the Beckmann-Spizzichino reflection model is based on physical optics analysis derived from the Maxwell's equations.

In comparison with diffuse reflections, in principle, specular reflections have three different properties [5]:

1. The diffuse and specular reflections have different degrees of polarization (DOP). The DOP represents the ratio of the light being polarized. For unpolarized incident light, the DOP of specular reflection is larger than that of diffuse reflection for most angles of incidence light, meaning that specular reflection is generally more polarized than diffuse reflection [6–8].
2. While recently a number of researchers (e.g., [9]) have introduced more complex models, the intensity distribution of diffuse reflections approximately follows Lambert's law [10]. In contrast, the intensity distribution of specular reflections generally follows the Torrance-Sparrow reflection model [1].

3. For optically inhomogeneous objects, the spectral power distribution (SPD) of specular reflection is determined by the object's interface spectral reflectance, which is mostly constant throughout the wavelength of visible spectrum, causing the SPD of specular reflections to be the same as the illumination's SPD [11]. In contrast, the SPD of diffuse reflection is determined by the object's body spectral reflectance. This spectral power distribution (color) independence of diffuse and specular reflections was described in the dichromatic reflection model proposed by Shafer [12].

Note that the condition that the SPD of specular reflections is the same as the illumination's SPD is called neutral interface reflection or NIR [11]. This mostly occurs for the surfaces of optically inhomogeneous objects (such as ceramics, plastics, paints); however, it does not always occur for the surfaces of optically homogeneous objects (such as gold, bronze, copper) [13].

Application

Many existing algorithms in computer vision assume perfect diffuse surfaces and deem specular reflections to be outliers. However, in the real world, the presence of specular reflections is inevitable since there are many objects that exhibit both diffuse and specular reflections. To properly acquire the diffuse-only reflections, a method to separate the two components robustly and accurately is required (e.g., [5], see Fig. 1 for an example of the separation). Once this separation has been accomplished, the specular reflection component can be advantageous since it conveys useful information of the object photometric properties (e.g., [6, 14]). Moreover, specular highlights are useful for estimating illumination colors or color constancy (e.g., [15]). Aside from the photometric properties, specular reflections can be employed to estimate the geometric properties (e.g., [16–18]).

Open Problems

Without knowing the polarizing states and only analyzing image intensities, specular detection from a single image is still an open problem.

References

1. Torrance K, Sparrow E (1966) Theory for off-specular reflection from roughened surfaces. *J Opt Soc Am* 57: 1105–1114
2. Nayar S, Ikeuchi K, Kanade T (1991) Surface reflection: physical and geometrical perspectives. *IEEE Trans Pattern Anal Mach Intell* 13(7):611–634
3. Cook R, Torrance K (1981) A reflectance model for computer graphics. *Comput Graph* 15:307–316
4. Beckmann P, Spizzochino A (1963) The scattering of electromagnetic waves from rough surfaces. Pergamon, New York
5. Tan RT, Ikeuchi K (2005) Separating reflection components of textured surfaces using a single image. *IEEE Trans Pattern Anal Mach Intell* 27(2):178–193
6. Miyazaki D, Tan RT, Hara K, Ikeuchi K (2003) Polarization-based inverse rendering from a single view. In: 9th IEEE international conference on computer vision (ICCV 2003), Nice, France, pp 982–987
7. Wolff LB (1990) Polarization-based material classification from specular reflection. *IEEE Trans Pattern Anal Mach Intell* 12(11):1059–1071
8. Nayar SK, Fang XS, Boulton T (1996) Separation of reflection components using color and polarization. *Int J Comput Vis* 21(3):163–186
9. Wolff LB, Nayar S, Oren M (1998) Improved diffuse reflection models for computer vision. *Int J Comput Vis* 30(1):55–71
10. Lambert JH (1760) *Photometria sive de mensura de gratibus luminis, colorum et umbrae. sumptibus viduae E. Klett*
11. Lee H, Breneman E, Schulte C (1990) Modeling light reflection for computer color vision. *IEEE Trans Pattern Anal Mach Intell* 12(4):402–409
12. Shafer S (1985) Using color to separate reflection components. *Color Res Appl* 10:210–218
13. Healey G (1989) Using color for geometry-insensitive segmentation. *J Opt Soc Am A* 6(6):920–937
14. Nishino K, Zhang Z, Ikeuchi K (2001) Determining reflectance parameters and illumination distribution from a sparse set of images for view-dependent image synthesis. In: 8th IEEE international conference on computer vision (ICCV 2001), Vancouver, Canada, pp 599–606
15. Tan RT, Nishino K, Ikeuchi K (2004) Color constancy through inverse-intensity chromaticity space. *J Opt Soc Am A* 21:2004
16. Blake A, Brelstaff G (1988) Geometry from specularities. In: 2nd IEEE international conference on computer vision (ICCV 1988), Tampa, USA, pp 394–403
17. Oren M, Nayar SK (1997) A theory of specular surface geometry. *Int J Comput Vis* 24(2):105–124
18. Adato Y, Vasilyev Y, Ben-Shahar O, Zickler T (2007) Toward a theory of shape from specular flow. In: 11th IEEE international conference on computer vision (ICCV 2007), Rio de Janeiro, Brazil, pp 1–8

Spherical Camera

► Omnidirectional Camera

Splines

Bo Zheng

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

Synonyms

Piecewise polynomial

Related Concepts

►Algebraic Curve; ►Parametric Curve

Definition

In mathematics, splines are piecewise continuous functions, such as polynomials, defined in successive subintervals. They are often used to represent one- or multidimensional data set (e.g., a curve or a surface) in the applications requiring interpolation, smoothing or nonrigid transformation [1]. For example, a spline curve is a piecewise collection of curve segments defined in polynomials that are connected end to end to form a single continuous curve. A curve in L -dimensional space can be simply defined by the following form:

$$S : [a, b] \rightarrow \mathbb{R}^L, \quad (1)$$

where function S takes variables from an interval $[a, b]$ and maps them to an L -dimensional real number. If the interval $[a, b]$ is divided into k ordered disjoint subintervals t_i, t_{i+1} with

$$a = t_0 \leq t_1 \leq \dots \leq t_k = b, i = 0, \dots, k-1, \quad (2)$$

then in each subinterval $[t_i, t_{i+1}]$ there is a polynomial defined as

$$P_i : [t_i, t_{i+1}] \rightarrow \mathbb{R}^L. \quad (3)$$

Therefore,

$$S(t) = P_i(t), t_i \leq t < t_{i+1}, \quad (4)$$

where t_i is called a *knot* and vector $\mathbf{t} = (t_0, \dots, t_k)^T$ is called a *knot vector*.

Many types of splines have been developed through that $P(t)$ is defined in different types of functions. Some examples of representative splines are Bézier spline, B-spline, Nonuniform rational B-spline and Thin-plate spline which are briefly introduced in the following sections.

Bézier Spline

The polynomial for Bézier spline of degree n is

$$P(t) = \sum_{i=0}^n B_i^n(t) \mathbf{p}_i, \quad (5)$$

where \mathbf{p}_i ($\in \mathbb{R}^L$) are called the control points of a Bézier spline, and $B_i^n(t)$ are the basis functions determined by *Bernstein polynomials* of degree n as

$$B_i^n(t) = \frac{n!}{i!(n-i)!} t^i (1-t)^{n-i}. \quad (6)$$

Several Bézier spline curves can be concatenated by sharing the first and last control points. While a Bézier spline has C^1 continuity to its defined interval, at the shared control points it gets C^0 continuity. C^1 continuity can be achieved by sharing two control points at the end of two curves. An improvement that adds an adjustable weight to each basis can make it easier to control and more closely approximated to arbitrary shapes.

B-Spline

Similar to Bézier spline, the polynomial for B-spline is defined as

$$P(t) = \sum_{i=0}^n N_{i,k}(t) \mathbf{p}_i, \quad (7)$$

where \mathbf{p}_i ($\in \mathbb{R}^L$) are called the control points of a B-spline curve and the basis functions $N_{i,k}(t)$ of degree k can be derived by the recurrence equations as

$$N_{i,1}(t) = \begin{cases} 1 & (t_i \leq t < t_{i+1}) \\ 0 & (\text{otherwise}) \end{cases} \quad (8)$$

$$N_{i,k}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1}(t) \quad (9)$$

The knots t_i for B-spline are generally with uniform, open uniform or nonuniform intervals.

Compared to Bézier splines, B-splines can have C^2 continuity at the joint control points. B-splines is more flexible and pleasing to work with, and thus it is popular in various graphics development environment.

Nonuniform Rational B-Splines (NURBS)

Since B-spline can be viewed as weighted sum of its control points and the weights $N_{i,k}$ usually have the property: $\sum_{i=0}^n N_{i,k}(t) = 1$. As weights $N_{i,k}(t)$ of B-spline depend only on the knot vector, it is useful to add to every control point one more weight w_i which can be set independently as

$$P(t) = \frac{\sum_{i=0}^n w_i N_{i,k}(t) \mathbf{p}_i}{\sum_{i=0}^n w_i N_{i,k}(t)}. \quad (10)$$

Then increasing a weight w_i makes the point more influential and attracts the curve to it. NURBS is often employed in computer-aided design systems.

Spline Surface

Not only for curve representation, spline is easy to be extended for representing a surface segment. Appropriately parameterized 2D variables are required for defining the 2D region subinterval to control a surface. For example, through a parameterized u - v plane, Bézier spline can be used to represent a surface with the following form:

$$P(u, v) = \sum_{i=0}^m \sum_{j=0}^n B_i^m(u) B_j^n(v) \mathbf{p}_{ij}, \quad (11)$$

where $\mathbf{p}_{ij} (\in \mathbb{R}^3)$ are the control points of Bézier spline surface.

Thin-Plate Spline (TPS)

Different to Bézier spline or B-spline in parametric form, Thin-plate spline (TPS) explicitly takes the point \mathbf{x} on a curve or a surface as the variable and maps it to a new value $f(\mathbf{x})$. A TPS function is often composed of a summation of radial basis functions and a low-order

polynomial, e.g., defined in the following form:

$$f(\mathbf{x}) = \sum_{i=1}^N w_i \phi(\|\mathbf{x} - \mathbf{c}_i\|) + \mathbf{v}^T \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix}, \quad (12)$$

where \mathbf{c}_i are the control points of TPS; w_i are the mapping coefficients through weighting the basis functions $\phi(\cdot)$; \mathbf{v} is the coefficient vector of the polynomial of degree 1; basis function $\phi(\cdot)$ is defined as

$$\phi(r) = \begin{cases} r^k & \text{with } k = 1, 3, 5, \dots, \\ r^k \ln(r) & \text{with } k = 2, 4, 6, \dots \end{cases} \quad (13)$$

where k is equal to the dimension of dataset.

TPS has been widely applied for building a smooth nonrigid transformation model, by minimizing the integral bending energy [4]. It is also useful for data interpolation, since it can explicitly represent the arbitrarily spaced tabulated data set, e.g. $(x_i, y_i, f(x_i, y_i))$ in 2D case. The interpolation is smooth with derivatives of any order.

Background

The idea of presenting a curve using the connection with splines comes from the ship building industry, where they construct templates for ships by passing thin strips of wood (called “splines”). In some subfields of computer science, wide class of spline functions are applied to the applications where the discrete data requires interpolation and/or smoothing. Because splines are superior in terms of the following qualities: the simplicity of their construction, the ease and accuracy of control, their capacity to approximate complex shapes and the ability to design curves interactively.

Application and Theory

In addition to the usages of interpreting or smoothing for image representation, such as the work in [9], splines also play an important role in some specific computer vision applications. For example, in the Snake-based image segmentation designed by Kass et al. [5] and modified by Brigger et al. [3], splines are used to model the image contours by minimizing

the energy under the guidance of external and internal forces; For image motion estimation, Szeliski and Coughlan [8] proposed to represent the local motion flow field using multi-resolution splines; A classical linear method for estimating the TPS coefficients for image warps is proposed by Bookstein [2]; And Free Form Deformation (FFD) proposed by Sederberg [7] represents the nonrigid deformation of object using grid B-splines, which has a successful application in medical image registration [6].

References

1. Bartels RH, Beatty JC, Barsky BA (1987) An introduction to splines for use in computer graphics and geometric modeling. Morgan Kaufmann, San Francisco
2. Bookstein FL (1989) Principal warps: thin-plate splines and the decomposition of deformations. *IEEE Trans Pattern Anal Mach Intell* 11:567–585
3. Brigger P, Hoeg J, Unser M (2000) B-spline snakes: a flexible tool for parametric contour detection. *IEEE Trans Image Process* 9(9):1484–1496
4. Duchon J (1977) Splines minimizing rotation-invariant seminorms in Sobolev spaces. *Constr Theory Funct Several Var Lect Notes Math* 571:85–100
5. Kass M, Witkin A, Terzopoulos D (1988) Snakes: active contour models. *Int J Comput Vis* 1:321–331
6. Rueckert D, Sonoda LI, Hayes C, Hill DLG, Leach MO, Hawkes DJ (1999) Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Trans Med Imaging* 18:712–721
7. Sederberg T (1986) Free-form deformation of solid geometric models. *ACM SIGGRAPH Comput Graph* 20(4):151–160
8. Szeliski R, Coughlan J (1997) Spline-based image registration. *Int J Comput Vis* 22(3):199–218
9. Unser M, Aldroubi A, Eden M (1991) Fast b-spline transforms for continuous image representation and interpolation. *IEEE Trans Pattern Anal Mach Intell* 13(3):277–285

Standard Illuminants

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

[CIE standard illuminant](#)

Related Concepts

► [Chromaticity](#)

Definition

A standard illuminant, as defined by the International Commission on Illumination (Commission Internationale de L'Eclairage, abbreviated as the CIE), denotes a specific relative distribution of energy (“spectral power distribution”) for an illuminant, over the range 300–830 nm [3].

Background

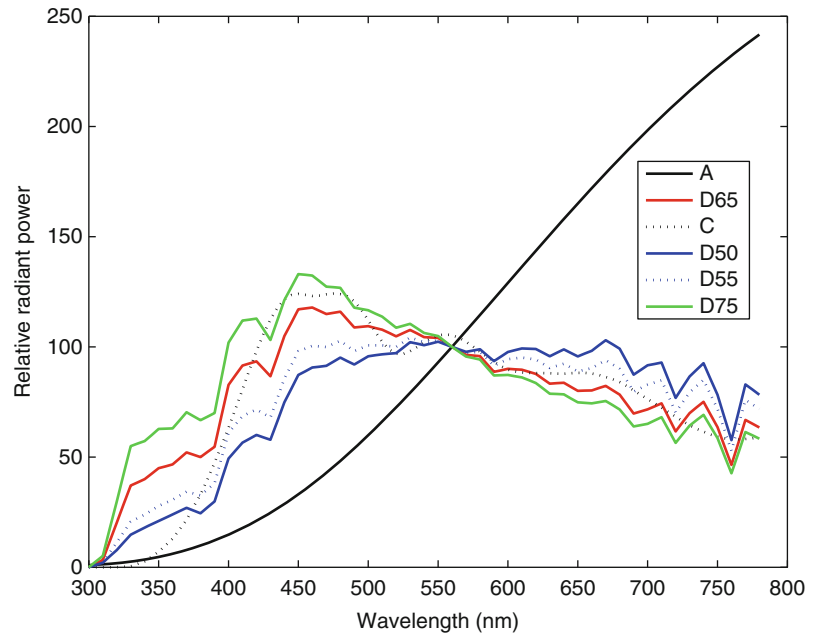
The motivation for a definition of standard illuminants arises from the need for accurate measurement of the colors of objects. It is well understood that for a Lambertian surface with a spectral reflectance function given by $r(\lambda)$, and under an illuminant given by $i(\lambda)$, the three-vector of CIE XYZ tristimulus values is given by [8]:

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

where \bar{x} , \bar{y} , \bar{z} denote the three CIE XYZ color-matching functions (based on a standard observer) and k is a constant. Working in the XYZ color space – natively three-dimensional in nature – when a color changes it is difficult to attribute the change to a specific function of wavelength, because of the above 3-D projection: the observer color-matching functions $\{\bar{x}, \bar{y}, \bar{z}\}$, the illuminant $i(\lambda)$, or the object's reflectance $r(\lambda)$. When the primary objective of a measurement is to quantify the object's reflectance properties, it is therefore helpful to use a standard observer and a standard illuminant. The standardization of the observer is well documented in various CIE standards since the early 1900s, most recently in CIE 15-2004 [1].

Standard Illuminants, Fig. 1

Standard illuminants as specified by the CIE, normalized to a peak of 100 at 560 nm

**Theory**

In CIE-1998c (and in CIE 15-2004), the CIE defines the following standard illuminants, along with their relative spectral power distribution, in the spectral range 300–830 nm [1, 3]:

- Standard Illuminant A
- Standard Illuminant C
- Standard Illuminant D50
- Standard Illuminant D55
- Standard Illuminant D65
- Standard Illuminant D75

The spectral range 300–830 nm, wider than the range of visual perception, which is approximately 360–830 nm ([8], p.122), is used specifically to enable the evaluation of luminescent samples where the ultraviolet range becomes important. For colorimetric measurements, however, typically 380–780 nm is used.

Standard Illuminant A represents a black-body radiator with a temperature of approximately 2,856 K. Standard Illuminant C, although not used often, represents average daylight with correlated color temperature of about 6,800 K. Standard Illuminant D65 is

used to represent a phase of daylight with a correlated color temperature of approximately 6,500 K and is by far the most commonly used standard illuminant in colorimetry. Other D-illuminants are standard daylight illuminants annotated with the first two digits of their correlated color temperature; e.g., D55 denotes a standard daylight illuminant with a correlated color temperature of 5,500 K. Figure 1 shows the relative spectral power distributions of the above standard illuminants.

It is to be noted that although the plots in Fig. 1 show the spectra in the extended range from 300 to 780 nm (as is given in the CIE standard), data from 300–330 nm to 780–830 nm is extrapolated, but considered sufficiently accurate for colorimetric purposes. Further, the standard provides data to 5 nm increments, and should finer increments be needed the standard recommends performing linear interpolation on the spectra.

The correlated color temperature (T) of a daylight illuminant, for the purposes of interpolating standard daylights, is related to its chromaticity coordinates in the x, y chromaticity diagram by the following equations ([6], p. 111):

$$y_D = -3.0000x_D^2 + 2.870x_D - 0.275 \quad (2)$$

$$x_D = \begin{cases} \frac{-4.6070 \times 10^9}{T^3} + \frac{2.9678 \times 10^6}{T^2} + \frac{0.09911 \times 10^3}{T} + 0.244063, & 4,000 \text{ K} \leq T \leq 7,000 \text{ K} \\ \frac{-2.0064 \times 10^9}{T^3} + \frac{1.9018 \times 10^6}{T^2} + \frac{0.24748 \times 10^3}{T} + 0.237040, & 7,000 \text{ K} \leq T \leq 25,000 \text{ K}. \end{cases} \quad (3)$$

Note that several other suggestions for a best CCT have been posited [4, 7].

Figure 2 shows the locus of chromaticities of standardized daylight illuminants, specifically denoting the locations D55, D65, and also, for reference, the location of illuminant A and the equi-energy point, E ($x = y = 0.33$) (all for the CIE 1931 2° observer).

The CIE also specifies equations to compute the relative spectral power distribution of other D-illuminants as a function of their x, y chromaticity coordinates.

$$S(\lambda) = S_0(\lambda) + M_1 S_1(\lambda) + M_2 S_2(\lambda), \quad (4)$$

where $S_0(\lambda)$, $S_1(\lambda)$, and $S_2(\lambda)$ are specified in the CIE standard [1] and plotted in Fig. 3. These may be

considered as the mean and the following two eigenvectors of the space of daylight illuminants [5].

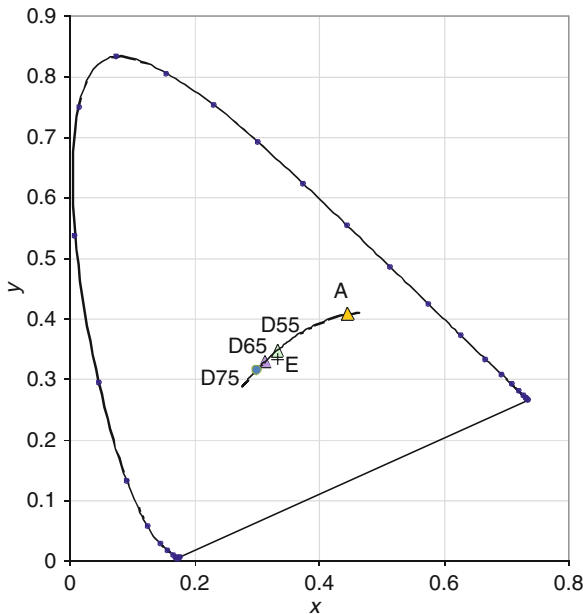
M_1 and M_2 are scale factors defined as a function of the chromaticity coordinate of illuminant, given by:

$$M_1 = \frac{-1.3515 - 1.7703x_D + 5.9114y_D}{0.0241 + 0.2562x_D - 0.7341y_D} \quad (5)$$

$$M_2 = \frac{0.03 - 31.4424x_D + 30.0717y_D}{0.0241 + 0.2562x_D - 0.7341y_D}. \quad (6)$$

These equations provides the relative spectral power distribution of the various D-illuminants, which are shown in Fig. 4. Customarily, $S(\lambda)$ is scaled such that its value at wavelength $\lambda = 560 \text{ nm}$ is 100.

It is perhaps most important to note that standard illuminants are not the same as standard sources – which are the real-world equivalents of standard illuminants. In other words, a standard illuminant – defined by its relative spectral power – may not necessarily be realizable by a physical emitter of radiation (see Fig. 5 for an example of a D65 source made by a fluorescent lamp and the D65 illuminant). The details of the challenges between the “theoretical” illuminant that is useful for computations and the “real” source may be found in a different CIE standard [2], and also in the book by Wyszecki and Stiles [8].



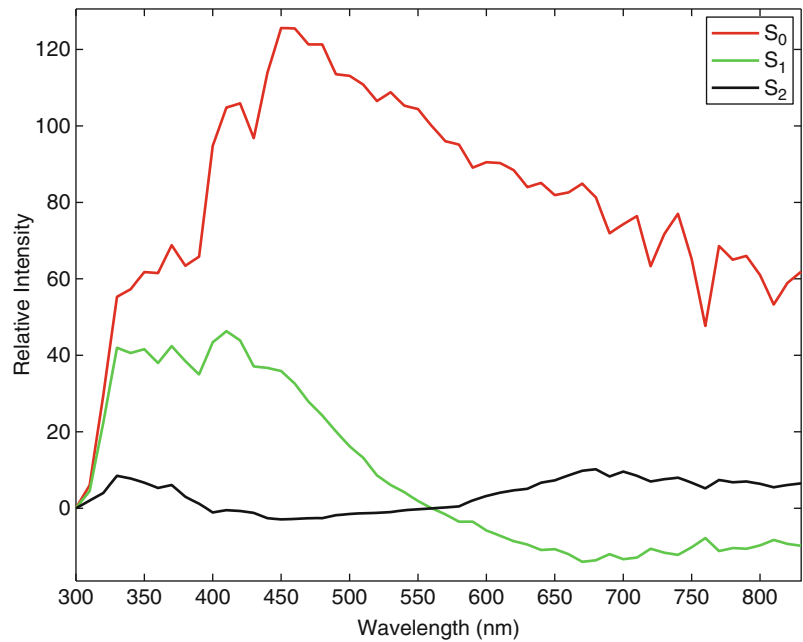
Standard Illuminants, Fig. 2 CIE x, y chromaticity diagram showing illuminants D55, D65, D75, A, and E (+, equi-energy illuminant)

References

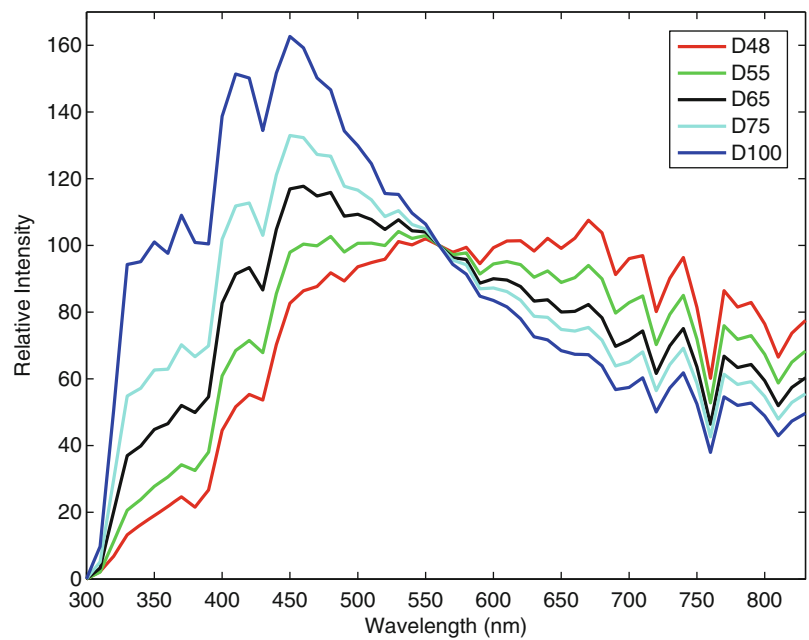
1. CIE 15:2004 (2004) Colorimetry. CIE, Vienna
2. CIE 51.2–1999 (1999) A method for assessing the quality of daylight simulators for colorimetry. CIE, Vienna
3. 1998c CIE (1998) CIE standard illuminants for colorimetry. CIE, Vienna. Also published as ISO 10526/CIE/S006/E1999
4. Hernández-Andrés J, Lee RL, Romero J (1999) Calculating correlated color temperatures across the entire gamut of daylight and skylight chromaticities. Appl Opt 38(27): 5703–5709

Standard Illuminants, Fig. 3

Component vectors (S_0 , S_1 , S_2 defined by the CIE), also considered as the mean and two additional eigenvectors of the space of daylight illuminants

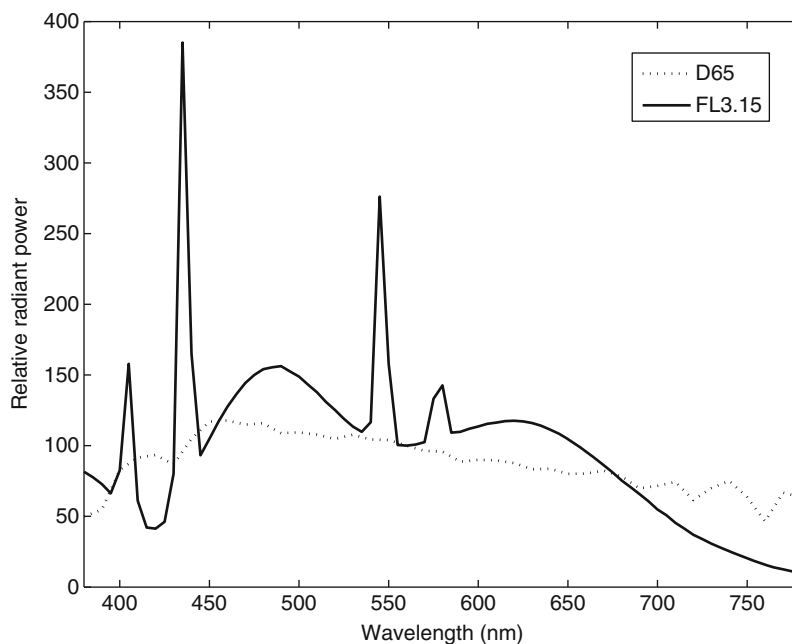
**Standard Illuminants, Fig. 4**

Daylight illuminants of different correlated color temperatures as computed from the component vectors (S_0 , S_1 , S_2 defined by the CIE)



Standard Illuminants, Fig. 5

Comparison of a standard illuminant (D65) and a standard source (FL 3.15) normalized to 100 at 560 nm



5. Judd DB, MacAdam DL, Wyszecki G, Budde HW, Condit HR, Henderson ST, Simonds JL (1964) Spectral distribution of typical daylight as a function of correlated color temperature. *J Opt Soc Am* 54:1031–1036
6. Judd DB, Wyszecki G (1975) *Color in business, science and industry*, 3rd edn. Wiley, New York
7. Ohno Y, Jergens M (1999) Results of the intercomparison of correlated color temperature calculation. Council for optical radiation measurements, June 16 1999. CORM subcommittee CR3 photometry. Boulder, Colorado
8. Wyszecki G, Stiles WS (1982) *Color science: concepts and methods, quantitative data and formulas*, 2nd edn. Wiley, New York

Definition

Statistical independence is a concept in probability theory. Two events A and B are statistical independent if and only if their joint probability can be factorized into their marginal probabilities, i.e., $P(A \cap B) = P(A)P(B)$. If two events A and B are statistical independent, then the conditional probability equals the marginal probability: $P(A|B) = P(A)$ and $P(B|A) = P(B)$. The concept can be generalized to more than two events. The events A_1, \dots, A_n are independent if and only if $P(\bigcap_{i=1}^n A_i) = \prod_{i=1}^n P(A_i)$.

Statistical Cooling

► [Simulated Annealing](#)

Statistical Independence

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

Related concepts

► [Principal Component Analysis \(PCA\)](#)

Theory

Two random variables X and Y are independent if and only if the events $\{X \leq x\}$ and $\{Y \leq y\}$ are independent for all x and y , that is, $F(x, y) = F_X(x)F_Y(y)$, where $F(x, y)$ is the joint cumulative distribution function and F_X and F_Y are the marginal cumulative distribution functions of X and Y , respectively. If X and Y are continuous random variables, then X and Y are independent if $f(x, y) = f_X(x)f_Y(y)$, where $f(x, y)$ is the joint probability density function and f_X and f_Y are the marginal probability density functions of X and Y , respectively. Similar results hold when both X

and Y are discrete, or one is discrete and the other is continuous.

If two random variables X and Y are independent, then their covariance $\text{Cov}(X, Y) = E(XY) - E(X)E(Y) = 0$, that is, they are uncorrelated. However, the reverse may not be true. Two uncorrelated random variables are not necessarily independent of each other. For example, if $X \sim \text{Uniform}[-1, 1]$ and $Y = X^2$, then $\text{Cov}(X, Y) = 0$, but clearly they are not independent.

The concept of independence can be generalized to more than two random variables. In probability theory, the law of large number, the central limit theorem, and concentration inequalities are usually obtained for independent random variables, although these results can be generalized to dependent random variables. In statistical learning theory, it is usually assumed that the training and testing examples are independent and identically distributed.

Application

Statistical independence is a key assumption in independent component analysis (ICA) [1], where the observed multivariate signal is assumed to be linear mixing of independent sources. A useful extension is independence is conditional independence. Two events A and B are independent given event C if $P(A \cap B|C) = P(A|C)P(B|C)$. If X , Y , and Z are discrete random variables, then X and Y are independent given Z if $P(X = x, Y = y|Z = z) = P(X = x|Z = z)P(Y = y|Z = z)$ for all x, y, z . If X , Y , and Z are continuous, then X and Y are independent given Z if $f_{X,Y|Z}(x, y|z) = f_{X|Z}(x|z)f_{Y|Z}(y|z)$, where $f_{X,Y|Z}$ is the conditional probability density function of (X, Y) given Z and $f_{X|Z}$ and $f_{Y|Z}$ are the conditional probability density functions of X given Z and Y given Z , respectively.

Conditional independence is a key assumption in Markov chains, Markov random fields, and more generally graphical models [2].

References

1. Hyvärinen A, Karhunen J, Oja E (2001) Independent component analysis. Wiley, New York
2. Cowell RG, Dawid AP, Lauritzen SL, Spiegelhalter DJ (1999) Probabilistic networks and expert systems. Springer, New York

Statistical Shape Analysis

Anuj Srivastava¹, Sebastian Kurtek²
and Eric Klassen²

¹Florida State University, Tallahassee, FL, USA

²Ohio State University, Columbus, OH, USA

Synonyms

[Morphology, form analysis](#)

Related Concepts

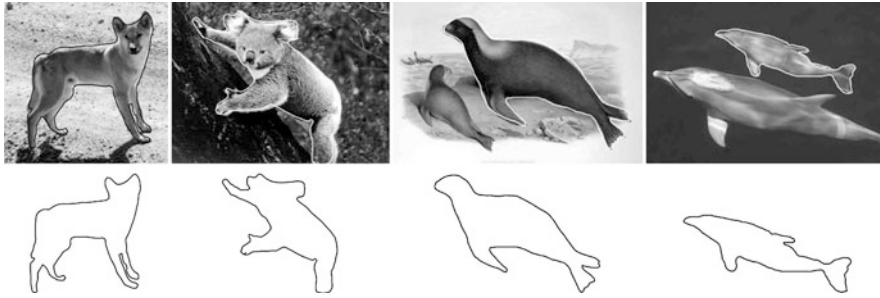
► [Activity Recognition](#); ► [Object Detection](#)

Definition

What is shape? Although the use of words *shape* or *shape analysis* is very common in computer vision, its definition is seldom made precise in a mathematical sense. According to the Oxford English Dictionary, it means “the external form or appearance of someone or something as produced by their outline.” Kendall [1] described shape as a mathematical property that remains unchanged under certain transformations such as rotation, translation, and global scaling. Shape analysis seeks to represent shapes as mathematical quantities, such as vectors or functions, that can be manipulated using appropriate rules and metrics. *Statistical* shape analysis is concerned with quantifying shape as a random quantity and developing tools for generating shape comparisons, averages, probability models, hypothesis tests, Bayesian estimates, and other statistical procedures on shape spaces.

Background

Shape is an important physical property of objects that characterizes their appearances and can play an important role in their detection, tracking, and recognition in images and videos. One usually restricts to the boundaries of objects, rather than the whole objects, for shape analysis and that leads to a shape analysis of curves (for 2D images) and surfaces (for 3D images).

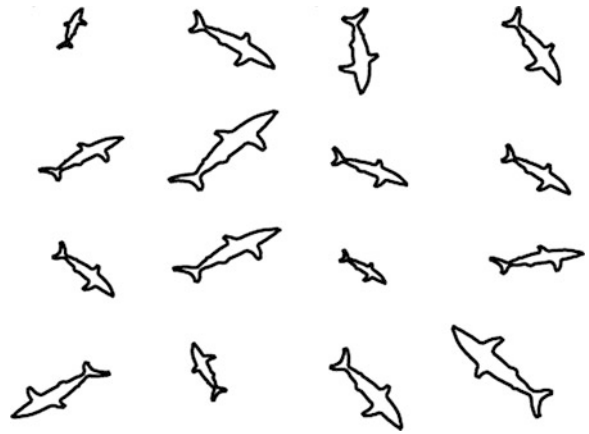


Statistical Shape Analysis, Fig. 1 Shapes of boundary curves are useful in object characterizations

Figure 1 suggests that shapes of boundaries can help characterize objects present in images. Any boundary contains only some partial information about the object since the color (texture) information inside and outside the boundary is lost. However, even with this limited information, it is often possible to broadly classify an object using shape analysis.

To understand the issues and challenges in shape analysis, one has to look at the imaging process since that is a major source of shape data. A picture can be taken from an arbitrary pose (arbitrary distance and orientation of the camera relative to the imaged object), and this introduces a random rotation, translation, and scaling of boundaries in the image plane. Therefore, any proper metric for shape analysis should be independent of the pose and scale of the boundaries. A visual inspection also confirms that any rotation, translation, or scaling of a boundary, while changing its coordinates, does not change its shape. Figure 2 shows an example of 16 curves that differ in orientations, scales, and locations, but still represent the same shape.

In case of parameterized curves and surfaces, an additional challenge arises when it comes to invariance. Let $\beta : [0, 1] \rightarrow \mathbb{R}^2$ represent a parameterized curve, and let $\gamma : [0, 1] \rightarrow [0, 1]$ be a smooth, invertible function such that $\gamma(0) = 0$ and $\gamma(1) = 1$. Then, the composition $\tilde{\beta}(t) \equiv (\beta \circ \gamma)(t)$ represents a curve with coordinate functions that are different from those of $\beta(t)$ but have the same shape. $\tilde{\beta}$ is called a *re-parameterization* of β . Figure 3 illustrates this issue with a simple example. It shows that the coordinate functions of the re-parameterized curve, $\tilde{\beta}_x(t)$ and $\tilde{\beta}_y(t)$, as functions of t , are different from the original coordinate functions $\beta_x(t)$ and $\beta_y(t)$. But when $\tilde{\beta}_x(t)$ is plotted versus $\tilde{\beta}_y(t)$, it traces out the same curve as that traced by $\beta_x(t)$ versus $\beta_y(t)$. This results in an

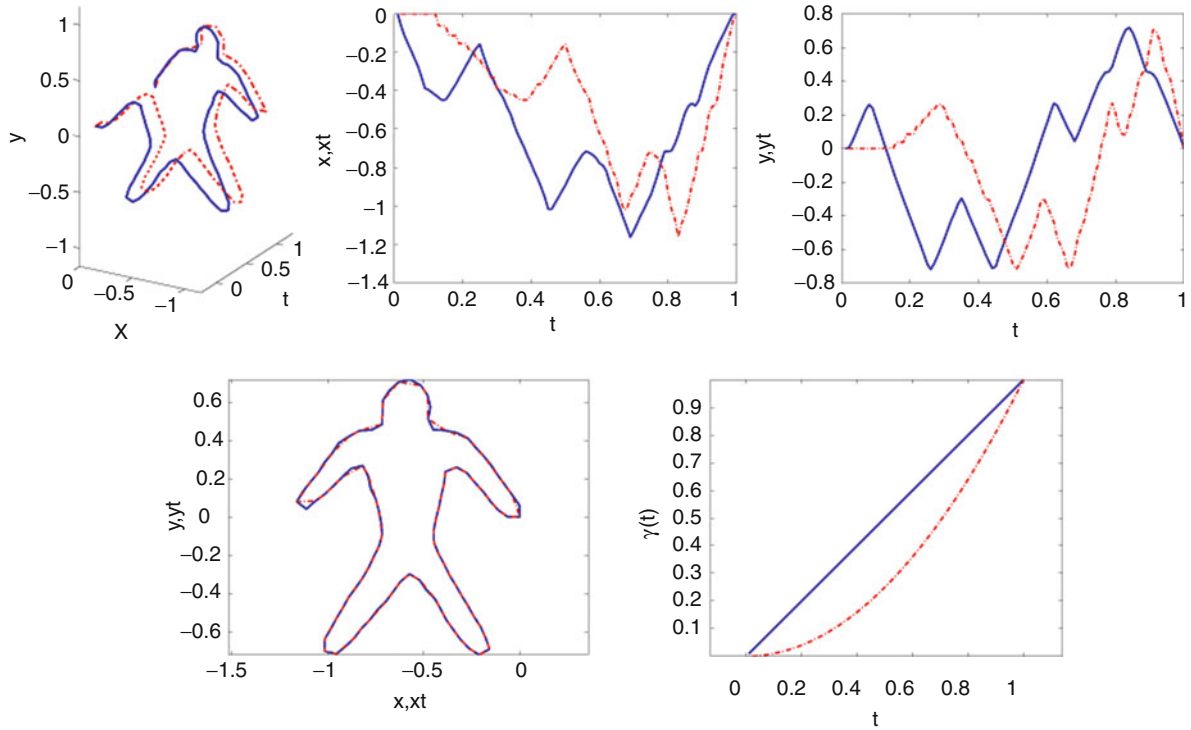


Statistical Shape Analysis, Fig. 2 Sixteen curves with different orientations, scale, and locations, but with identical shapes

additional invariance requirement in shape analysis of parameterized curves (and similarly for surfaces). That is, the shape metrics should be invariant to how the curves are parameterized.

In *statistical* shape analysis, one treats shapes as random quantities and tries to answer questions of the type:

- What is the difference between shapes of any two given objects? How can such shape differences be quantified while maintaining the desired invariances?
- What shape best represents the shapes of a given collection of objects? Another way to ask the same questions is: What is the statistical average of a given collection of shapes?
- What are the principal modes of variations in a given set of shapes?
- How can one capture the main shape variability in a population using probability models? Can random shapes be generated from such models?



Statistical Shape Analysis, Fig. 3 Re-parameterized curve has different coordinate functions but same shape as the original curve. *Left:* curves $(t, \beta_x(t), \beta_y(t))$ and $(t, \tilde{\beta}_x(t), \tilde{\beta}_y(t))$. Next

two panels: $\beta_x(t)$ and $\tilde{\beta}_x(t)$, and $\beta_y(t)$ and $\tilde{\beta}_y(t)$. *Right:* curves $(\beta_x(t), \beta_y(t))$ and $(\tilde{\beta}_x(t), \tilde{\beta}_y(t))$

- How can one use such probability models for shape classification and object recognition?

To start answering such questions, one needs precise mathematical representations of shapes and tools from algebra and geometry for analyzing them. In the literature there are numerous mathematical representations of objects that have been used for this purpose. These include representations using (unordered) point sets, (ordered) landmarks, level sets, deformable images, medial representations, and parameterized curves and surfaces. An important aspect, common to most representations, is the nonlinear geometry of a shape space. It is easy to recognize that shape is not a quantity that can be added, averaged, or grouped easily using Euclidean calculus. The desired invariance of shape to certain transformations (rigid motion, global scaling, and re-parameterization) implies that some nontraditional tools are needed.

Among the different methods used in shape analysis, the earlier methods typically represented objects by

finite sets of points, such as point sets or landmarks, but more recent methods are beginning to handle parameterized curves and surfaces directly as functions. Two of the earlier ideas based on point-set representations are summarized next.

Active Shape Models (ASM)

The active shape models approach to shape analysis was introduced by Cootes and Taylor in [2]. The simplest idea in shape analysis is to sample the boundaries at a number of points and form polygonal shapes by connecting those points with straight lines. Of course, the number and locations of these points on the objects can drastically change the resulting polygonal shapes, but this issue will be disregarded for the moment. One can organize the coordinates of these points in a form of a vector of coordinates and perform standard vector calculus. Let $x \in \mathbb{R}^{n \times 2}$ represent n ordered points selected from the boundary of an object. It is often convenient to identify points in \mathbb{R}^2 with elements of \mathbb{C} , i.e., $x^i \equiv z^i = (x^{i,1} + jx^{i,2})$, where $j = \sqrt{-1}$. Thus, in this

complex representation, a configuration of n points x is now $z \in \mathbb{C}^n$. Before analyzing the shape of z , it is “standardized” by moving its center to the origin (of the coordinate system):

$$z^i \mapsto \left(z^i - \frac{1}{n} \sum_{i=1}^n z^i \right).$$

To remove the scale variability, z is rescaled to have norm one, i.e., $z \mapsto z/\|z\|$. Then, one uses tools from standard multivariate statistics to analyze and model them. So far, the translation and the scale variability of a configuration are removed, but the rotation remains. That is, two configurations, z and a rotation of z , will have a nonzero distance between them even when they have the same shape. This problem is solved by using an additional step of rotational alignment when comparing shapes, as follows:

$$\begin{aligned} \phi^* &= \operatorname{argmin}_{\phi \in \mathbb{S}^1} \|z_1 - e^{j\phi} z_2\|^2 \\ &= \operatorname{argmin}_{\phi \in \mathbb{S}^1} (\|z_1\|^2 + \|e^{j\phi} z_2\|^2 - 2\Re(\langle z_1, e^{j\phi} z_2 \rangle)) \\ &= \operatorname{argmax}_{\phi \in \mathbb{S}^1} (\Re(e^{-j\phi} \langle z_1, z_2 \rangle)) = \theta, \\ &\text{where } \langle z_1, z_2 \rangle = r e^{j\theta}, \end{aligned} \quad (1)$$

and $\langle \cdot, \cdot \rangle$ is the standard Hermitian inner product. The distance between the two configurations is then $\|z_1 - e^{j\phi^*} z_2\| = \sqrt{2(1-r)}$. The corresponding optimal deformation from one shape to another is simply a straight line between z_1 and $e^{j\phi^*} z_2$, i.e., $\alpha_{asm}(\tau) = (1-\tau)z_1 + \tau e^{j\phi^*} z_2$ for $\tau \in [0, 1]$.

One remaining issue in this analysis is that on a closed curve which point should be selected as z^1 , the first point or the *seed*. If there are n points sampled on a curve, then there are n candidates for the seed. The solution is to select the best seed during a pairwise comparison of configurations. That is, select any point on the first configuration as the seed for the first shape and try all n points in the second configuration as candidate seeds for the second shape. Of those, select the one that results in the smallest distance from the first configuration. Figure 5 shows several examples of these deformations: one between a pair of human silhouettes, one between a pair of hands, and so on. These geodesics have been computed using $n = 200$ points on each configuration so that the resulting polygons look like smooth curves.

One can define the mean shape of several configurations z_1, z_2, \dots, z_k as the configuration that minimizes the sum of squares of distances:

$$\mu_{asm} = \operatorname{argmin}_{z \in \mathbb{C}^n} \sum_{i=1}^k \|z - e^{j\phi_i^*} z_i\|, \quad \phi_i^* = \cos^{-1}(\Re(\langle z, z_i \rangle)).$$

Figure 6 shows several illustrations of the mean computations.

Although this technique is relatively simple and fast, it has some important limitations. One limitation is that it does not preserve the scale constraints imposed on the shape representations. For instance, the intermediate shapes along the optimal deformations between any two unit length configurations do not have unit length. Similarly, the mean shape of $\{z_i\}$ is generally not of unit length. This is because no effort is made to restrict to the set of unit-length configurations, a limitation that is addressed in the next approach.

Kendall's Shape Analysis (KSA)

This approach, first laid out by Kendall [1] and advanced by several others [3], preserves desired constraints by restricting to appropriate manifolds. Once again a configuration of n points, taken from the boundary of an object, is treated as a complex vector. As earlier, the translations are removed by restricting to those elements of \mathbb{C}^n whose average is zero and the scale variability by rescaling the complex vector to have norm one. This results in a set:

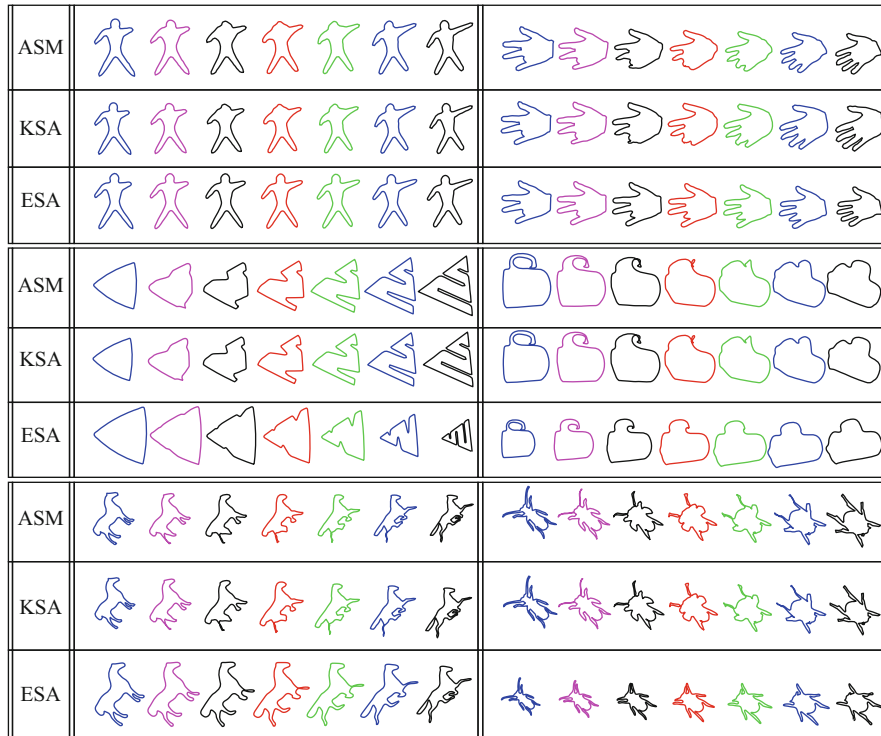
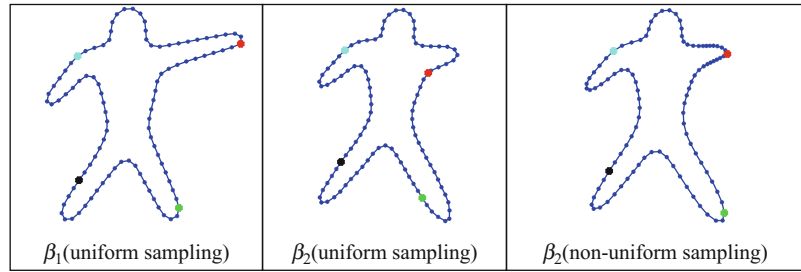
$$\mathcal{D} = \left\{ z \in \mathbb{C}^n \mid \frac{1}{n} \sum_{i=1}^n z^i = 0, \|z\| = 1 \right\}.$$

\mathcal{D} is not a vector space because $a_1 z_1 + a_2 z_2$ for $a_1, a_2 \in \mathbb{R}$ and $z_1, z_2 \in \mathcal{D}$ is typically not in \mathcal{D} , due to the unit norm constraint. However, \mathcal{D} is a unit sphere and one can utilize the geometry of a sphere to analyze points on it. Under the Euclidean metric, the shortest path between any two elements $z_1, z_2 \in \mathcal{D}$, also called a *geodesic*, is given by the great circle: $\alpha_{ksa} : [0, 1] \rightarrow \mathcal{D}$, where

$$\begin{aligned} \alpha_{ksa}(\tau) &= \frac{1}{\sin(\theta)} [\sin(\theta(1-\tau))z_1 + \sin(\tau\theta)z_2], \quad \text{and} \\ \theta &= \cos^{-1}(\Re(\langle z_1, z_2 \rangle)). \end{aligned} \quad (2)$$

Statistical Shape

Analysis, Fig. 4 Registration of points across two curves using the uniform and a convenient nonuniform sampling. Nonuniform sampling allows a better matching of features between β_1 and β_2



Statistical Shape Analysis, Fig. 5 Examples of geodesic paths between same shapes using ASM, KSA, and ESA

In order to compare the shapes represented by z_1 and z_2 , they need to be aligned rotationally, as was done earlier, but the shape space is defined more formally this time. Let $[z]$ be the set of all rotations of a configuration z according to:

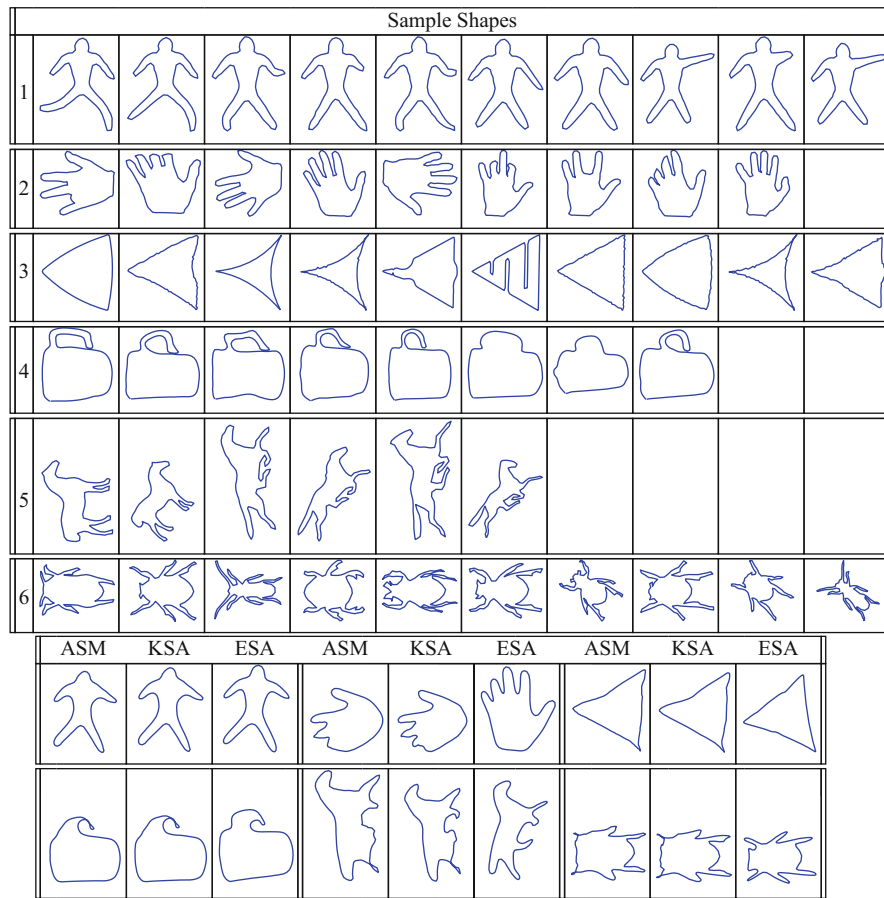
$$[z] = \{e^{j\phi}z | \phi \in \mathbb{S}^1\}.$$

One defines an equivalence relation on \mathcal{D} by setting all elements of this set as equivalent, i.e., $z_1 \sim z_2$ if there exists an angle ϕ such that $z_1 = e^{j\phi}z_2$. The set of all such equivalence classes is the quotient space $\mathcal{D}/U(1)$, where $U(1) = SO(2) = \mathbb{S}^1$ is the set of all rotations

in \mathbb{R}^2 . This space is called the *complex projective space* and is denoted by \mathbb{CP}^{n-1} . A geodesic between two elements $z_1, z_2 \in \mathbb{CP}^{n-1}$ is given by computing α_{ksa} between z_1 and $e^{j\phi^*}z_2$, where ϕ^* is the optimal rotational alignment of z_2 to z_1 . The length of the geodesic is given by θ and that quantifies the difference in shapes of the boundaries represented by z_1 and z_2 . [Figure 5](#) shows several examples of geodesic paths between the same shapes as for the ASM examples.

Issue of Landmark Selection

Although Kendall's approach succeeds in preserving the unit-length constraints on the landmark



Statistical Shape Analysis, Fig. 6 Examples of mean shapes under three different methods

configurations, it does not address a very important practical issue: How to systematically select points on objects, say curves, to form representative point sets? This process is difficult to standardize and different selections can lead to drastically differing solutions. This issue is present in any point-based approach, including the ASM method discussed above. Some may suggest to sample a curve uniformly along its length, i.e., parameterize a curve β using arc length and sample $\{\beta(t_i)|i = 0, 1, 2, \dots, n\}$ where $t_i = i/n$. Although this provides a standardized way of sampling curves, the results are not always good since this forces a particular registration of points, i.e., the point $\beta_1(t_i)$ on the first curve is matched to the point $\beta_2(t_i)$ on the second curve, irrespective of the shapes involved. Figure 4 illustrates this point using an example. Shown in the left two panels are two curves: β_1 and β_2 , sampled uniformly along their lengths. For

$t_i = i/4$, $i = 1, 2, 3, 4$ the corresponding four points on each curve $\{\beta_1(t_i)\}$ and $\{\beta_2(t_i)\}$ are shown in the same color. While two of the four pairs seem to match well, the pairs shown in red and green fall on different parts of the body, resulting in a mismatch of features. This example shows the pitfall of using uniform sampling of curves. In fact, any predetermined sampling and preregistration of points will, in general, be problematic. A more natural solution is to treat the boundaries of objects as *continuous* curves, rather than discretize them into point sets at the outset, and find an optimal (perhaps nonuniform) sampling, such as the one shown in the rightmost panel, that better matches features across curves. This way one can develop a more comprehensive solution, including theory and algorithms, assuming continuous objects and will discretize them only at the implementation stage.

Theory

The *elastic shape analysis* (ESA) framework, for analyzing shapes represented by simple, closed, planar, curves is described here. (These ideas are also applicable, with some modifications, to curves in higher dimensions and to shape analysis of surfaces, but that is not discussed here.) The most natural way to study shapes of curves seems to be by treating them as parameterized curves. As mentioned earlier, an important aspect of this framework is that shape distances, geodesics, and statistics should be invariant to how the curves are parameterized.

Basic Challenge

To understand the basic challenge in analyzing shapes of parametrized curves, let $\|\cdot\|$ denote the \mathbb{L}^2 norm of a vector-valued function, i.e., $\|\beta\| = \int_{\mathbb{S}^1} \|\beta(t)\|^2 dt$, where the norm inside the integral is the vector 2-norm. Let $\beta_1, \beta_2 : \mathbb{S}^1 \rightarrow \mathbb{R}^2$ be two parameterized closed curves and γ be a re-parameterization function of the type used in Fig. 3. (It is more natural for the domain of parameterization to be \mathbb{S}^1 instead of $[0, 1]$ for closed curves.) The basic challenge in using the \mathbb{L}^2 norm $\|\beta_1 - \beta_2\|$ for comparing shapes of these two curves, even after a proper translation and scaling for standardization, is that $\|\beta_1 - \beta_2\| \neq \|\beta_1 \circ \gamma - \beta_2 \circ \gamma\|$ in general. That is, the distance between any two curves changes even if they are re-parameterized in the same way. This implies that the shape distance will depend on parameterizations, and this violates the requirement of invariance to parameterization. This problem necessitates a new representation and/or a new metric for analyzing shapes of curves.

Mathematical Representation

Let a parameterized closed curve be denoted as $\beta : \mathbb{S}^1 \rightarrow \mathbb{R}^2$. In order to analyze its shape, one represents β by its square-root velocity function (SRVF): $q(t) = \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}} \in \mathbb{R}^2$. The SRVF q includes both the instantaneous speed ($\|q(t)\|^2 = \|\dot{\beta}(t)\|$) and the direction ($\frac{q(t)}{\|q(t)\|} = \frac{\dot{\beta}(t)}{\|\dot{\beta}(t)\|}$) of the curve β at time t . The use of the time derivative makes the SRVF invariant to any translation of curve β . Conversely, one can reconstruct the curve β from q up to a translation. In order for the shape analysis to be invariant to scale, one rescales each curve to length one. With a slight abuse of notation, let us denote the rescaled curves by β . Since

$\int_{\mathbb{S}^1} \|\dot{\beta}(t)\| dt = 1$: $\int_{\mathbb{S}^1} \|q(t)\|^2 dt = \int_{\mathbb{S}^1} \|\dot{\beta}(t)\| dt = 1$. In other words, the \mathbb{L}^2 norm of the SRVF q is one. Additionally, if the curve β is closed, then its SRVF satisfies $\int_{\mathbb{S}^1} q(t) \|q(t)\| dt = 0$. Restricting to the curves of interest, represented by their SRVFs, the following set is obtained:

$$\mathcal{C} = \{q : \mathbb{S}^1 \rightarrow \mathbb{R}^2 \mid \int_{\mathbb{S}^1} q(t) \|q(t)\| dt = 0, \int_{\mathbb{S}^1} \|q(t)\|^2 dt = 1\}.$$

\mathcal{C} is called the *preshape space* and is the set of SRVFs of all unit length, closed curves in \mathbb{R}^2 . Four shape-preserving transformations were mentioned: translation, scale, rotation, and re-parameterization. Of these, the first two have already been eliminated from the representations, but the other two remain. Curves that are within a rotation and/or a re-parameterization of each other result in different elements of \mathcal{C} despite having the same shape. The unification of such curves is performed algebraically as follows: Let $SO(2)$ be the group of 2×2 rotation matrices and Γ be the group of all re-parameterizations (they are actually positive diffeomorphisms of the unit circle \mathbb{S}^1). For a curve β , a rotation $O \in SO(2)$ and a re-parameterization $\gamma \in \Gamma$, the transformed curve is given by $O(\beta \circ \gamma)$. The SRVF of the transformed curve is given by $O(q \circ \gamma) \sqrt{\dot{\gamma}}$. In order to unify all elements in \mathcal{C} that denote the same shape, one can define equivalence classes of the type:

$$[q] = \{O(q \circ \gamma) \sqrt{\dot{\gamma}} \mid O \in SO(2), \gamma \in \Gamma\}.$$

Each such equivalence class $[q]$ is associated with a shape uniquely and vice versa. The set of all these equivalence classes is called the *shape space* \mathcal{S} ; mathematically, it is a quotient space of the preshape space: $\mathcal{S} = \mathcal{C}/(SO(2) \times \Gamma)$. The preshape space \mathcal{C} is a nonlinear manifold because it is a subset of a unit sphere. One cannot perform calculus on this space as if it is a vector space. Operations such as addition, subtraction, and multiplication are not available on nonlinear spaces. This means that standard techniques in functional analysis for inferences on \mathcal{C} and \mathcal{S} cannot be used.

It should be noted that the mathematical representation used here, i.e., the SRVF, is not the only such representation. Younes et al. [4, 5] use a slightly different expression for an SRVF, based on an identification

of \mathbb{R}^2 with \mathbb{C} and taking complex square roots of the coordinates. The advantage of the SRVF form used in the current article is that it applies to curves in \mathbb{R}^m for any m , while the complex analysis in [4, 5] is applicable only to curves in \mathbb{R}^2 .

Elastic Riemannian Metric

Now that a mathematical representation of the shape of a curve, namely $[q]$, has been defined, how should one compare the shapes of two curves? In other words, for any two curves β_1 and β_2 , represented by their shape classes $[q_1]$ and $[q_2]$, respectively, what should be the shape metric $d_s([q_1], [q_2])$ that quantifies their shape differences? First, consider the role of a metric more closely. When one curve is deformed into another, a continuous sequence of curves, or a path in the curve space, is generated, and a natural question is how long that path is. The length of this path also quantifies the amount of deformation in going from one curve to the other. The question changes to the following: What should be the metric to measure this path length? A metric called the *elastic metric* will be used for this purpose. An elastic metric is a metric that measures the amount of bending and stretching between successive curves along the path and adds them up for the full path. Mio et al. [6] defined a family of elastic metrics depending upon how much relative weights are attached to bending and stretching. More recently, it was shown that (Joshi et al. [7], Srivastava et al. [8]) under the SRVF representation, the complicated elastic metric turns into (using a change of variables) the standard \mathbb{L}^2 metric. That is, one can alternatively compute the path lengths, or the sizes of deformations between curves, using the cumulative norms of the differences between successive curves along the paths in the SRVF space. This turns out to be much simpler and a very effective strategy for comparing shapes of curves, by finding the paths with least amounts of deformations between them, where the amount of deformation is measured by an elastic metric. Another distinct advantage of using SRVFs is that for any $q_1, q_2 \in \mathcal{C}$, $O \in SO(2)$ and $\gamma \in \Gamma$:

$$\|q_1 - q_2\| = \|O(q_1 \circ \gamma)\sqrt{\dot{\gamma}} - O(q_2 \circ \gamma)\sqrt{\dot{\gamma}}\|.$$

This means that the distance between any two curves remains the same if they are rotated and reparameterized in the same way! This property, when

combined with an optimization step Eq. 3 (later), allows one to make shape metrics invariant to parameterizations.

Shape Comparison and Geodesics

Once a Riemannian metric on a manifold has been defined, one can compute distances between points in that manifold. For any two points, the distance between them is given by the length of the shortest path, called a *geodesic*, connecting them in that manifold. An interesting feature of this framework is that it not only provides a distance between shapes of two curves but also a geodesic path between them in \mathcal{S} . This path has the interpretation that it provides the optimal deformation of one shape into another. The geodesics are actually computed using the differential geometry of the underlying space \mathcal{S} . Consider two curves β_1 and β_2 , represented by their SRVFs q_1 and q_2 . Let $\alpha : [0, 1] \rightarrow \mathcal{C}$ be a differentiable path connecting them in \mathcal{C} . The length of this path is given by

$$L[\alpha] = \int_0^1 \langle \dot{\alpha}(\tau), \dot{\alpha}(\tau) \rangle^{1/2} d\tau,$$

where the inner product inside the integral is given by the elastic Riemannian metric. A geodesic is a path whose length cannot be minimized by locally perturbing it. It is often obtained by minimizing the cost function of the type:

$$\hat{\alpha} = \underset{\{\alpha : [0,1] \rightarrow \mathcal{C} | \alpha(0)=q_1, \alpha(1)=q_2\}}{\operatorname{argmin}} \left(\int_0^1 \langle \dot{\alpha}(\tau), \dot{\alpha}(\tau) \rangle dt \right).$$

This cost function differs from the expression for $L[\alpha]$ in that the square root inside the integral is missing. It can be shown that a local minimizer of this cost function is also a local minimizer of $L[\alpha]$ and, hence, is a geodesic.

One technique for finding geodesics is called *path straightening*. It is an iterative technique that initializes an arbitrary path and then iteratively “straightens” it by updating it along the negative gradient of the cost function. Klassen and Srivastava [9] provide a nice analytical expression for the gradient of this cost function that results in a convenient gradient iteration. One applies these iterative updates, or straightening, until the gradient becomes negligible and the resulting path is the desired geodesic $\hat{\alpha}$. The length of this curve is denoted by $d_c(q_1, q_2) = L[\hat{\alpha}]$. This gives a geodesic

and a geodesic distance between SRVFs in \mathcal{C} , but the goal is to compute geodesic paths in \mathcal{S} . In other words, geodesic paths between the equivalence classes $[q_1]$ and $[q_2]$ are needed and not just q_1 and q_2 . It turns out that the desired geodesic is obtained by finding the shortest geodesic among all pairs $(\tilde{q}_1, \tilde{q}_2) \in ([q_1] \times [q_2])$. This search is further simplified by fixing an arbitrary element of $[q_1]$, say q_1 , and searching over all rotations and re-parameterizations of q_2 to minimize the geodesic length:

$$(O^*, \gamma^*) = \operatorname{argmin}_{O \in SO(2), \gamma \in \Gamma} d_c(q_1, O(q_2 \circ \gamma) \sqrt{\dot{\gamma}}). \quad (3)$$

The minimization over $SO(2)$ is similar to the ASM and KSA alignments earlier, but the optimization over Γ is new. This is accomplished using the dynamic programming algorithm or a gradient-type approach [8]. The resulting geodesic between q_1 and $O^*(q_2 \circ \gamma^*) \sqrt{\dot{\gamma}^*}$ is actually a geodesic between $[q_1]$ and $[q_2]$ in the shape space \mathcal{S} ; its length $d_s([q_1], [q_2]) = d_c(q_1, O^*(q_2 \circ \gamma^*) \sqrt{\dot{\gamma}^*})$ provides the geodesic distance in \mathcal{S} . Several examples of geodesic paths in the shape space \mathcal{S} are shown in Fig. 5; these geodesics are compared with the deformations/geodesics obtained by two previously described methods: ASM and KSA, for the same shapes. It is easy to see that the geodesics resulting from ESA provide more natural deformations as they are better in matching features across shapes.

The role of geodesics is preeminent in this framework because (1) its length d_s is a quantitative measure of difference between shapes of curves represented by q_1 and q_2 ; (2) this measure is invariant to rigid motion, scaling, and re-parameterization of any of the curves; (3) it also provides a full deformation (along the geodesic path) for taking one shape into another in an optimal way; and (4) the availability of geodesics leads to a further development of tools for building statistical summaries of shape classes, as described next.

Mean Shape

The first important task in statistical shape analysis is to define and compute the mean shape for a set of curves. Compared to the sample means of real-valued random variables, this task is not straightforward since the shape space \mathcal{S} is not a vector space. One cannot simply take the SRVFs of the given curves and average them point by point to get a mean shape. The notion

of a mean on a nonlinear manifold is typically established using the Karcher mean [10]. For a given set of curves $\beta_1, \beta_2, \dots, \beta_n$, represented by their SRVFs q_1, q_2, \dots, q_n , their Karcher mean is defined as the quantity that satisfies:

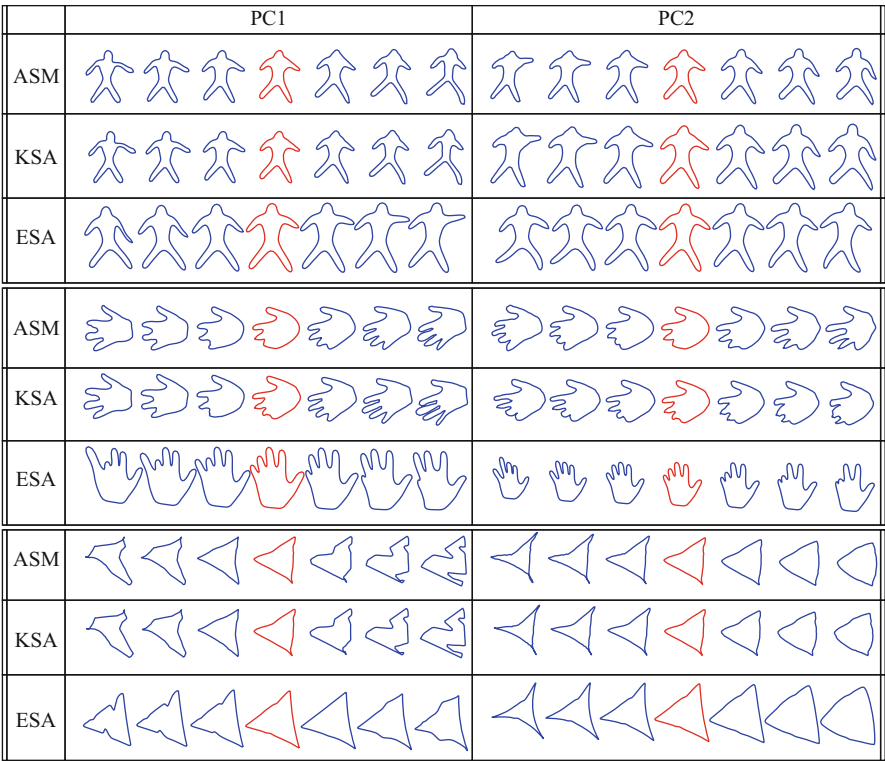
$$[\mu] = \operatorname{argmin}_{[q] \in \mathcal{S}} \sum_{i=1}^n d_s([q], [q_i])^2.$$

There is a gradient-based iterative algorithm for finding the minimizer of this cost function that can be found in [10–12]. Since this algorithm is based on a local search, the solution obtained is usually local and depends on the initial condition. Shown in Fig. 6 are some examples of mean shapes. The top six rows show a set of given curves and bottom rows display their means computed using the three methods discussed here: ASM, KSA, and ESA.

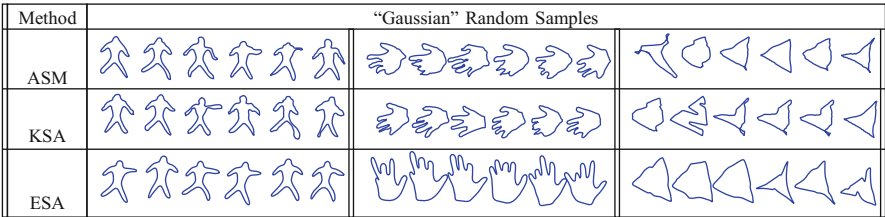
Shape Covariance and Principal Modes of Variation

Now that the first moment, i.e., the mean, of a set of curves has been defined, one can look for the higher moments. The role of the second centralized moment, the covariance, is especially important as (1) one can define a Gaussian distribution using just the mean and the covariance and (2) the singular value decomposition (SVD) of the covariance matrix can be used for a principal component analysis (PCA) of shape data. These two ideas are briefly summarized next, starting with the PCA.

For computing and analyzing the second and higher moments of a shape sample, the tangent space to the shape manifold \mathcal{S} at the point μ is used. This space, denoted by $T_\mu(\mathcal{S})$, is convenient because it is a vector space and one can apply more traditional methods here. First, for each given curve q_i , the vector $v_i \in T_\mu(\mathcal{S})$ is computed such that a geodesic that goes from μ to q_i in unit time has the initial velocity v_i . The function $v_i : \mathbb{S}^1 \rightarrow \mathbb{R}^2$ is also called the *shooting vector* from μ to q_i . Let \hat{K} be the sample covariance matrix of all the shooting vectors from μ to the SRVFs q_i s. For capturing the essential variability in a shape set, one can use principal component analysis (PCA) of the shooting vectors. The basic idea is to compute the SVD $\hat{K} = U \Sigma U^T$, where U is an orthogonal matrix and Σ is the diagonal matrix of singular values. Assuming that the entries along the diagonal



Statistical Shape Analysis, Fig. 7 Two principal directions of variability in shapes given in Examples 1–3 in Fig. 6



Statistical Shape Analysis, Fig. 8 Random samples from the “Gaussian”-type distributions under the three methods: ASM, KSA, and ESA, for parameters estimated from the given shapes shown in Examples 1–3 of Fig. 6

in Σ are organized in a nonincreasing order, U_1 , U_2 , etc. represent the dominant directions of variability in the data. If a singular vector U_j is used to form vectors $t\sqrt{\Sigma_{jj}}U_j$, then they represent shooting variability in the direction U_j . One can map these directions back on the shape space \mathcal{S} using an exponential map at μ . The details of this exponential map are omitted here, but it is basically the point reached on \mathcal{S} by constructing a geodesic in the shooting direction so that the length of the geodesic is the same as the length of the shooting vector. The resulting geodesics are also called the *principal geodesic paths*. Figure 7 shows the

principal geodesic paths along U_1 and U_2 , respectively, for $t = -1.5$ to $t = 1.5$. Of course, the middle points in each row are the mean shape μ .

Probabilistic Shape Models

One important use of means and covariances of shape families is in devising “Gaussian”-type probability densities on the shape space \mathcal{S} . In case of ASM this idea is straightforward since the shape representations are simple vectors, and one can define multivariate normal densities on these vector spaces. However, for KSA and ESA the shape spaces are nonlinear

manifolds and such probability densities are not easy to define. One common idea to tackle the nonlinearity of the shape space is to impose a Gaussian distribution on the tangent space $T_\mu(S)$ since that is a vector space. In case of ESA this space is infinite dimensional, so the Gaussian model is actually imposed on a finite-dimensional subspace, e.g., a principal subspace, of $T_\mu(S)$. Let $\{U_j\}$, $j = 1, 2, \dots, k$ denote the singular vectors of the sample covariance matrix as earlier. Then, one can define a random vector $v \equiv \sum_{j=1}^k f_j U_j$ where $f_j \sim N(0, \Sigma_{jj})$ and define $q = \exp_\mu(v)$, the exponential map of v from $T_\mu(S)$ to the shape space S . The procedure defines a generative random model on the shape space and is easy to sample from. Shown in Fig. 8 are examples of random samples from S using means and covariances estimated from the given data in Examples 1–3 from Fig. 6. For comparison, this figure also shows random samples from similar Gaussian models but using ASM and KSA. It is easy to observe the superiority of the results obtained using ESA; the modeling results from ASM are typically the worst of the three methods.

Open Problems

Although there has been a significant progress in shape analysis of curves, especially the planar curves, several important problems remain open. Firstly, the choice of Gaussian-type models for capturing shape variability of curves is more for convenience than data driven. It is important to explore and develop statistical models on shape manifolds that are both efficient, e.g., parametric models are more efficient, and provide a better representation of the observed variability. The second family of open problems relates to the shape analysis of surfaces. One needs to develop special mathematical representations of parameterized surfaces that will enable their shape analysis in a manner that is invariant to their parameterizations. Some preliminary ideas in that direction have been proposed in [13].

References

1. Kendall DG (1984) Shape manifolds, procrustean metrics and complex projective spaces. *Bull Lond Math Soc* 16: 81–121
2. Cootes TF, Taylor CJ, Cooper DH, Graham J (1995) Active shape models: their training and application. *Comput Vis Image Underst* 61:38–59
3. Dryden IL, Mardia K (1998) Statistical shape analysis. Wiley, Chichester/New York
4. Younes L (1998) Computable elastic distance between shapes. *SIAM J Appl Math* 58(2):565–586
5. Younes L, Michor PW, Shah J, Mumford D, Lincei R (2008) A metric on shape space with explicit geodesics. *Mat E Appl* 19(1):25–57
6. Mio W, Srivastava A, Joshi SH (2007) On shape of plane elastic curves. *Intl J Comput Vis* 73(3):307–324
7. Joshi SH, Klassen E, Srivastava A, Jermyn IH (2007) A novel representation for riemannian analysis of elastic curves in \mathbb{R}^n . In: *Proceedings of IEEE conference on computer vision pattern recognition (CVPR)*, Minneapolis, pp 1–7
8. Srivastava A, Klassen E, Joshi SH, Jermyn IH (2010) Shape analysis of elastic curves in euclidean spaces. *IEEE Trans Pattern Anal Mach Intell* (Accepted for publication). doi:10.1109/TPAMI.2010.184
9. Klassen E, Srivastava A (2006) Geodesics between 3D closed curves using path-straightening. In: *Proceedings of European conference on computer vision (ECCV)* Graz. *Lecture notes in computer science* I, pp 95–106
10. Karcher H (1977) Riemannian center of mass and mollifier smoothing. *Commun Pure Appl Math* 30(5):509–541
11. Le HL, Kendall DG (1993) The riemannian structure of euclidean shape spaces: a novel environment for statistics. *Ann Stat* 21(3):1225–1271
12. Srivastava A, Joshi SH, Mio W, Liu X (2005) Statistical shape analysis: clustering, learning and testing. *IEEE Trans Pattern Anal Mach Intell* 27(4):590–602
13. Kurtsek S, Klassen E, Ding Z, Jacobson S, Jacobson J, Avison M, and Srivastava A (2011) Parameterization-invariant shape comparisons of anatomical surfaces. *IEEE Trans Med Imaging* 30(3):849–858

Stochastic Differential Equations in Infinite Dimensions

► Stochastic Partial Differential Equations

Stochastic Partial Differential Equations

Annika Lang

Seminar für Angewandte Mathematik, ETH Zürich,
Zürich, Switzerland

Synonyms

SPDE; Stochastic differential equations in infinite dimensions

Definition

A stochastic partial differential equation (SPDE) is a partial differential equation (PDE) with an extra stochastic term, e.g., an Itô integral. Sometimes partial differential equations, where the differential operator or the initial condition is disturbed, are also called SPDEs, but the more common term for these equations is random PDEs.

Background

First results on SPDEs and infinite-dimensional stochastic differential equations (SDEs) appeared in the mid-1960s. Ample publications and results are due to the end 1970s and early 1980s. Here the work by Walsh [20] and Pardoux [16] should be mentioned. In the early 1990s, Da Prato and Zabczyk published their book with an infinite-dimensional approach to SPDEs driven by Wiener processes [6]. In the last years, a number of books on SPDEs were published, in particular an extension of [6] by Peszat and Zabczyk [17]. Other recent publications to be mentioned are Chow [3], Prévôt and Röckner [18], and Holden et al. [8].

The motivation to study SPDEs was driven on the one hand from the internal development of analysis and theory of stochastic processes and on the other side from applications. Random phenomena studied in natural sciences needed to be described. Especially applications in physics, chemistry, biology, control theory, nonlinear filtering, engineering, and finance pushed the development of the theory of SPDEs and are still pushing it. In recent years, the applications also inspire the design of numerical methods to “create numbers,” i.e., to simulate the equations.

Theory

Let $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t \geq 0}, \mathbb{P})$ be a filtered probability space. The filtration is supposed to be right continuous, and \mathcal{F}_0 contains all \mathbb{P} -zero sets. A *stochastic differential equation* is given by

$$dX(t) = a(t, X(t)) dt + b(t, X(t)) dM(t) \quad (1)$$

with initial condition $X(0) = X_0$ and $X : \mathbb{R}_+ \times \Omega \rightarrow \mathbb{R}^d$. The initial condition might be a random

variable. This notation is the abbreviation for the integral equation

$$X(t) = X_0 + \int_0^t a(s, X(s)) ds + \int_0^t b(s, X(s)) dM(s).$$

In this notation, M is a stochastic process adapted to (\mathcal{F}_t) which is, e.g., a local martingale. In many applications, the stochastic process is a Brownian motion also called Wiener process and abbreviated by B or W . Especially in recent years, other typical stochastic processes that are considered are Lévy processes. The expression

$$\int_0^t b(s, X(s)) dM(s)$$

is a stochastic integral of Itô type (cf. [5, 9, 14, 15, 19]). In order to ensure the existence of a solution to Eq. (1), the functions $a : \mathbb{R}_+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, $b : \mathbb{R}_+ \times \mathbb{R}^d \rightarrow \mathbb{R}^d$, and the stochastic process M have to satisfy certain conditions. One possibility is that a and b are of linear growth and Lipschitz type.

One approach to derive an SPDE is to extend Eq. (1) to maps $X : \mathbb{R}_+ \times \Omega \rightarrow \mathbb{R}^d$ where d tends to infinity. Another possibility to derive SPDEs is to start with a PDE. Therefore, let A be a differential operator on a domain $D \subset \mathbb{R}^d$. Then,

$$\frac{\partial}{\partial t} u(t, x) = Au(t, x) + f(u(t, x)) + g(u(t, x)) \dot{\eta}(t, x)$$

with initial condition $u(0, x) = u_0(x)$ for $x \in D$ denotes an SPDE where $\dot{\eta}$ is white noise. As in general stochastic processes are almost surely nowhere differentiable with respect to time t , this notation is used seldom. Instead of that, SPDEs are integral equations, i.e.,

$$u(t, x) = u_0(x) + \int_0^t (Au(s, x) + f(u(s))) ds + \int_0^t g(u(s, x)) dM(s, x).$$

One approach to solve equations of that type was introduced by Da Prato and Zabczyk in 1992 [6]. There, a more general framework is used. Let H and U be separable Hilbert spaces, e.g., $L^2(D), \mathbb{R}^d$, or Sobolev

spaces $H^\alpha(D)$, then the previous equation can be rewritten as

$$u(t) = u_0 + \int_0^t (Au(s) + F(u(s))) ds + \int_0^t G(u(s)) dM(s),$$

where $F : D(F) \subset H \times \Omega \rightarrow H$, $G : D(G) \subset H \rightarrow L(U, H)$, and M is a U -valued stochastic process, e.g., a square integrable martingale like a Wiener or Lévy process. Here $D(F)$ denotes the domain of F , and $L(U, H)$ is the space of linear operators from U into H . The stochastic process u is a mapping from $\mathbb{R}_+ \times \Omega$ into the Hilbert space H . The abbreviated form of the previous equation is

$$du(t) = (Au(t) + F(u(t))) dt + G(u(t)) dM(t) \quad (2)$$

with $u(0) = u_0$. So instead of solving an SPDE, here Hilbert space-valued SDEs are solved. Existence and uniqueness of solutions might be shown, using, e.g., semigroup theory and classical SDE theory. In the theory of SPDEs, there exist three main concepts of solutions which are similar to those known from PDE theory:

1. *Strong solutions* [6]: A predictable H -valued stochastic process $u = (u(t), t \in \mathbb{R}_+)$ is a *strong solution* to Eq. (2), if:
 - For all $t \geq 0$, $u(t)$ takes values in $D(A) \cap D(F) \cap D(G)$ \mathbb{P} -a.s.
 - For all $t \geq 0$, it holds \mathbb{P} -a.s.

$$u(t) = u_0 + \int_0^t (Au(s) + F(u(s))) ds + \int_0^t G(u(s)) dM(s).$$

2. *Weak solutions* [17]: A predictable H -valued process u is a *weak solution* to Eq. (2), if:

- $\sup_{t \in [0, T]} \mathbb{E}(\|u(t)\|_H^2) < +\infty$ for all $T \in \mathbb{R}_+$
- For all $a \in D(A^*)$, $t \geq 0$, it holds \mathbb{P} -a.s.

$$\begin{aligned} \langle a, u(t) \rangle_H &= \langle a, u_0 \rangle_H + \int_0^t (\langle A^* a, u(s) \rangle_H \\ &\quad + \langle a, F(u(s)) \rangle_H) ds \\ &\quad + \int_0^t \langle G^*(u(s))a, dM(s) \rangle_{\mathcal{H}}. \end{aligned}$$

Here A^* denotes the adjoint operator and \mathcal{H} is the reproducing kernel Hilbert space generated by M .

3. *Mild solutions* [17]: Let A be the generator of a strongly continuous semigroup $(S(t), t \geq 0)$, then the stochastic process u is a *mild solution* to Eq. (2), if:

- $\sup_{t \in [0, T]} \mathbb{E}(\|u(t)\|_H^2) < +\infty$ for all $T \in \mathbb{R}_+$
- For all $t \geq 0$, it holds \mathbb{P} -a.s.

$$u(t) = S(t)u_0 + \int_0^t S(t-s)F(u(s)) ds + \int_0^t S(t-s)G(u(s)) dM(s).$$

Furthermore, there are viscosity solutions [13]. One could also think of solutions in probability or in expectation and many other concepts.

If A is the generator of a strongly continuous semigroup, similarly to SDE theory, solutions exist under linear growth and Lipschitz conditions, but many extensions are possible. The reader is referred to the literature for explicit conditions for existence and uniqueness results. In this context, an overview can be found in [17], but the subject is still evolving such that recent research papers will give more general results.

For applications, a suitable type of solution has to be chosen. This leads to the next section where some possible applications are discussed as well as the problem that solutions for most SPDEs are not known. Therefore, numerics of SPDEs have become more and more important within the last year.

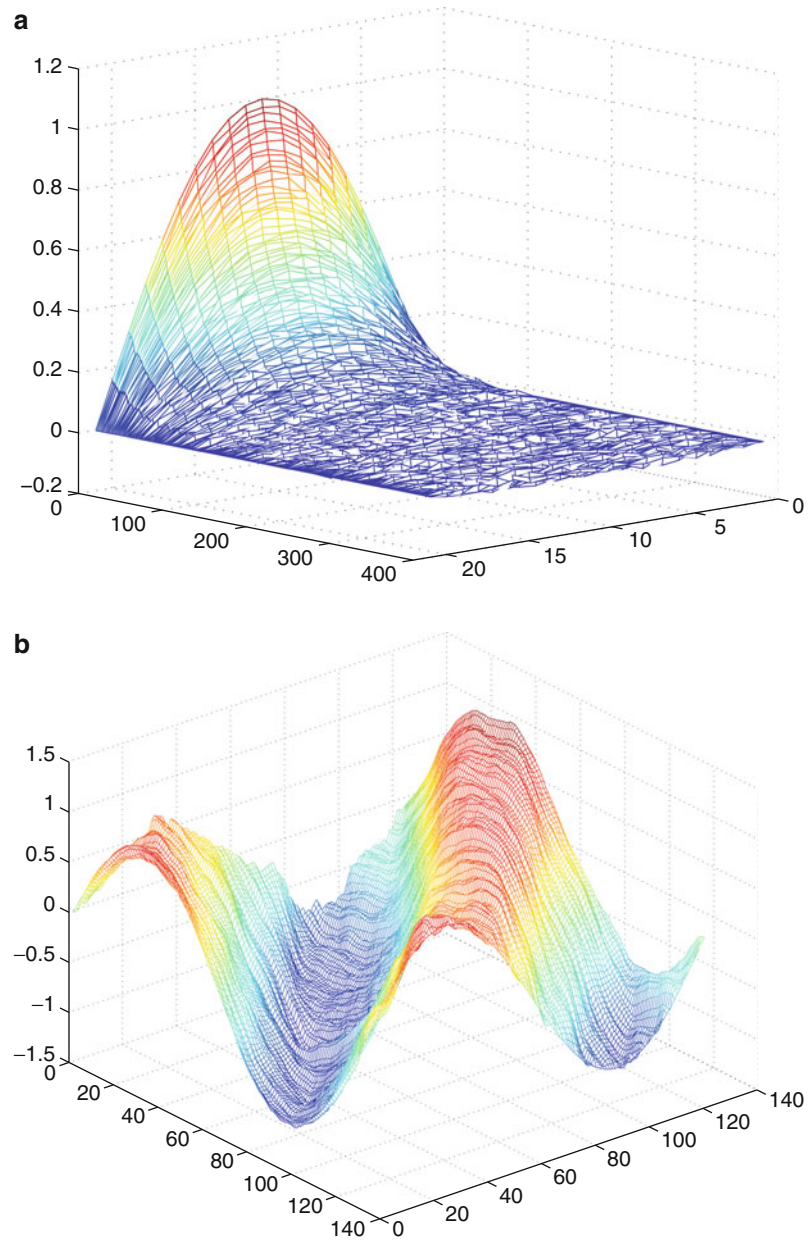
Application

SPDEs are relevant in many different applications. In engineering such as image analysis, surface analysis, and filtering, they result from addition of noise to PDEs. In finance, systems of SDEs are common that extend to infinite-dimensional problems and therefore to SPDEs. Furthermore, first applications to life and bio-sciences are done.

All these applications are interested in solutions of SPDEs, but these are not known explicitly in most cases. Therefore, numerics of SPDEs have become

Stochastic Partial Differential Equations, Fig. 1

Simulation of one path of an SPDE with additive noise on an interval. **(a)** Parabolic equation. **(b)** Hyperbolic equation



important within the last years. Methods combine SDE methods with PDE methods. Simulation methods for SDEs are especially Euler–Maruyama, Milstein, and higher-order schemes, where a good survey is given in [11], as well as Monte Carlo methods [7]. From PDE theory Galerkin methods, especially finite elements are relevant, where the reader is referred, e.g., to [4] and [2].

Open Problems

As the topic is fairly new, there are still many open questions. People are still working on existence and uniqueness theory and extending known results. Especially the simulation of SPDEs, i.e., the numerics evolve a lot in recent years, and so far, no book on the topic is available.



Stochastic Partial Differential Equations, Fig. 2 Segmentation using an SPDE and stochastic active contours. (a) initial condition. (b) Intermediate step. (c) Segmentation result

Experimental Results

To illustrate what an SPDE is and where to use it for, two figures are included. Fig. 1 shows two simulated paths, i.e., $\omega \in \Omega$ was chosen and a finite element method in space and an Euler–Maruyama scheme in time were used. On the left-hand side, the heat equation with additive noise and Dirichlet boundary conditions, i.e.,

$$du(t) = \Delta u(t) dt + dW(t)$$

on the space and time interval $[0, 1]$ with initial condition $u(0, x) = \sin(\pi x)$, is displayed, where W is a Wiener process with space correlation given by the kernel function $q(x, y) = \exp(-2|x - y|)$. On the right-hand side, a hyperbolic equation with additive noise is presented. It is given by

$$du(t) = \nabla u(t) dt + dW(t)$$

with initial condition $u(0, x) = \sin(2\pi x)$ and inflow boundary condition $u(t, 0) = -\sin(2\pi t)$. The other parameters are the same as for the heat equation. In both figures, time evolves from left to right. This and more simulation and convergence results can be found in [1].

One example for an application of SPDEs in computer vision is segmentation. This approach was suggested by Juan et al. [10]. Using level set methods and variational calculus, we obtain as one possible SPDE to be simulated

$$du(t, x) = \nabla \frac{\nabla u(t, x)}{|\nabla u(t, x)|} dt + |\nabla u(t, x)| dW(t, x),$$

where the initial condition is, e.g., a weighted distance function as in Fig. 2a. The red circles are the zero level sets. Different types of noise than the coupling with the size of the gradient can be found in [10] and [12]. An example of the segmentation of a zebra from [12] is shown in Fig. 2.

References

1. Barth A, Lang A (2012) Simulation of stochastic partial differential equations using finite element methods. *Stochastics* 84(2–3):217–231
2. Braess D (2007) *Finite elements. Theory, fast solvers and applications in elasticity theory* (Finite Elemente. Theorie, schnelle Löser und Anwendungen in der Elastizitätstheorie.) 4th rev and extended edn. Springer, Berlin
3. Chow P-L (2007) *Stochastic partial differential equations. Applied mathematics and nonlinear science series*. Chapman & Hall/CRC, Boca Raton
4. Dautray R, Lions J-L (2000) *Mathematical analysis and numerical methods for science and technology*. Springer, Berlin
5. Dellacherie C, Meyer P-A (1978) *Probabilities and potential* (Transl. from the French.) North-Holland mathematics studies, vol 29. North-Holland, Amsterdam/New York/Oxford
6. Da Prato G, Zabczyk J (1992) *Stochastic equations in infinite dimensions. Encyclopedia of mathematics and its applications*, vol 44. Cambridge University Press, Cambridge
7. Fishman GS (1996) *Monte Carlo. Concepts, algorithms, and applications*. Springer, New York
8. Holden H, Øksendal B, Ubøe J, Zhang T (2010) *Stochastic partial differential equations. A modeling, white noise functional approach*. Universitext, 2nd edn. Springer, New York
9. Ikeda N, Watanabe S (1989) *Stochastic differential equations and diffusion processes*. North-Holland

- mathematical library, 2nd edn., vol 24. North-Holland, Amsterdam/Kodansha, Tokyo
10. Juan O, Keriven R, Postelnicu G (2006) Stochastic motion and the level set method in computer vision: stochastic active contours. *Int J Comput Vis* 69(1):7–25
 11. Kloeden PE, Platen E (1992) Numerical solution of stochastic differential equations. *Applications of mathematics*, vol 23. Springer, Berlin
 12. Lang A (2007) Simulation of stochastic partial differential equations and stochastic active contours. PhD thesis, Universität Mannheim
 13. Lions P-L, Souganidis PE (1998) Fully nonlinear stochastic partial differential equations. *C R Acad Sci Paris Sér I Math* 326(9):1085–1092
 14. Métivier M (1982) Semimartingales: a course on stochastic processes. *de Gruyter studies in mathematics*, vol 2. Walter de Gruyter, Berlin/New York
 15. Øksendal B (2003) Stochastic differential equations. An introduction with applications. *Universitext*, 6th edn. Springer, Berlin
 16. Pardoux E (1979) Stochastic partial differential equations and filtering of diffusion processes. *Stochastics* 3:127–167
 17. Peszat S, Zabczyk J (2007) Stochastic partial differential equations with Lévy noise. An evolution equation approach. *Encyclopedia of mathematics and its applications*, vol 113. Cambridge University Press, Cambridge
 18. Prévôt C, Röckner M (2007) A concise course on stochastic partial differential equations. *Lecture notes in mathematics*, vol 1905. Springer, Berlin
 19. Protter PE (2004) Stochastic integration and differential equations. *Applications of mathematics*, 2nd edn., vol 21. Springer, Berlin
 20. Walsh JB (1986) An introduction to stochastic partial differential equations (*École d'été de probabilités de Saint-Flour XIV - 1984*.) *Lect Notes Math* 1180:265–437

Stochastic Relaxation

► Simulated Annealing

Structure-from-Motion (SfM)

► Factorization

Subpixel Estimation

Robert B. Fisher
School of Informatics, University of Edinburgh,
Edinburgh, UK

Synonyms

Superresolution

Definition

Subpixel estimation is the process of estimating the value of a geometric quantity to better than pixel resolution even though the data was originally sampled on an integer pixel quantized space.

Background

It is naively assumed that information at a scale smaller than the pixel level is lost when continuous data is sampled or quantized into pixels from, e.g., time-varying signals, images, data volumes, and space-time volumes. However, in fact, it may be possible to estimate geometric quantities to better than the original pixel accuracy. The underlying foundations of this estimation are the following:

- *Models of expected spatial variation*: Discrete structures, such as edges or lines, produce characteristic patterns of data when measured, allowing fitting of a model to the data to estimate the parameters of the structure.
- *Spatial integration during sampling*: Sensors typically integrate a continuous signal over a finite domain (space or time), leading to measurements whose values depend on the relative position of the sampling window and the original structure.
- *Point-spread function*: Knowledge of the PSF could be used, e.g., by deconvolution of a blurred signal, to estimate the position of the signal.

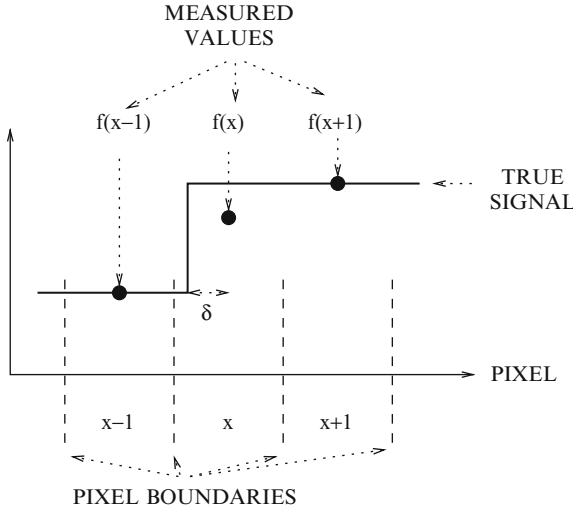
Applications commonly benefitting from subpixel estimation are (1) camera calibration and triangulation (e.g., in stereo and structured light depth estimation) and (2) image motion estimation for improved image stabilization and compression.

One of the earliest instances of subpixel edge detection in computer vision research was by MacVicar-Whelan and Binford [13] in 1981.

The accuracy of subpixel estimation depends on a number of factors, such as the image point-spread function, noise levels, and spatial frequency of the image data. A commonly quoted rule of thumb is 0.1 pixel, but lower is achievable, e.g., about 0.02 pixel is shown for stripe position detection in [1].

Theory

There are four common approaches to estimating subpixel positions:



Subpixel Estimation, Fig. 1 The values of $f(x)$ created by integrating the continuous signal over the whole pixel

1. *Interpolation:* An example is in subpixel edge position estimation, which is demonstrated here in one dimension in ideal form in Fig. 1. One can see that intensity $f(x)$ is a function of the edge's actual position within a pixel and the values at adjacent pixels. Here we assume that the pixel "position" refers to the center of the pixel. Let δ be the offset of the true edge position away from the pixel center. Then, one can model the value $f(x)$ at x in terms of the values at the neighbors, assuming a step function:

$$f(x) = \left(\frac{1}{2} + \delta\right) * f(x-1) + \left(\frac{1}{2} - \delta\right) * f(x+1)$$

from which we can solve for the subpixel edge position $x + \delta$ by:

$$\delta = \frac{2f(x) - f(x-1) - f(x+1)}{2(f(x-1) - f(x+1))}$$

Another approach is to interpolate a continuous curve (or surface) and then find the optimal position on the reconstructed curve (e.g., by using correlation for curve registration).

2. *Integration:* An example is the estimation of the center point of a circular dot, such as required for control point localization in a camera calibration scheme. The assumption is that the minor deviations from many boundary pixels can be accumulated to give a more robust estimate. Suppose that $g(x, y)$

are the gray levels of a light circle on a dark background, where (x, y) are in a neighborhood N closely centered on the circle. Assume also that the mean dark background level has been subtracted from all values. Then, the center of the dot is estimated by its gray-level center of mass:

$$\hat{x} = \frac{\sum_{(x,y) \in N} xg(x, y)}{\sum_{(x,y) \in N} g(x, y)}$$

and similarly for \hat{y} .

3. *Taylor series approximation:* An example is the subpixel feature point position estimation in the SIFT [12] operator. Given the difference of Gaussian function $D(x)$, where x represents the two spatial and one scale dimensions, the Taylor series expansion is:

$$D(x + \delta) = D(x) + \frac{\partial D(x)}{\partial x} \delta + \frac{1}{2} \delta^T \frac{\partial^2 D(x)}{\partial x^2} \delta$$

Differentiating with respect to δ and setting to 0 give the subpixel (and subscale) estimate:

$$\delta = -\frac{\partial^2 D(x)}{\partial x^2}^{-1} \frac{\partial D(x)}{\partial x}$$

4. *Phase correlation:* The key principle behind phase correlation is the assumption that the pattern of data across a whole window is more distinctive than the individual pixel values. The technique is also independent of intensity, so it can be used for multispectral or illumination-varying registration. Assume that we have two image windows f_a and f_b and their discrete Fourier transforms $F_a = \mathcal{F}(f_a)$ and $F_b = \mathcal{F}(f_b)$. Compute the cross-power spectrum as $F_a F_b^*$ (by elementwise multiplication) where $*$ is the complex conjugate, normalize elementwise by $|F_a F_a^*|$, and finally apply the inverse Fourier transform:

$$T = \mathcal{F}^{-1} \left(\frac{F_a F_b^*}{|F_a F_a^*|} \right)$$

The peak position in T is the desired offset. For subpixel alignment, the above method can be used to remove the integer component of the registration. Thereafter, one can estimate the subpixel peak position of the original registration or repeat the

process on an upsampled version of the image windows once the integer portion of the offset has been removed.

Application

Subpixel methods have been developed to analyze the following:

- *Shape parameters*: circle and other “blob” shape parameters [9], ellipse parameters for improved camera calibration [20], photometric stereo [19], superresolution [18], decomposition of mixed pixels formed by imaging two or more source types [2]
- *Feature positions*: point-like signals [10], “interest” points [12], “edge” transitions [15], “line” transitions [6]
- *Shape matching and registration*: image registration using phase analysis [7, 16] or spatial domain matching [11], motion estimation prior to image compression [17], stereo matching [8] and disparity estimation [14], feature tracking [3], optical flow [4], image and video stabilization [5]

References

1. Alexander BF, Ng KC (1991) Elimination of systematic error in subpixel accuracy centroid estimation. *Opt Eng* 30(9):1320–1331
2. Bovolo F, Bruzzone L, Carlin L (2010) A novel technique for subpixel image classification based on support vector machine. *IEEE Trans Image Process* 19(11):2983–2999
3. Brantner S, Auer T, Pinz A (1999) Real-time optical edge and corner tracking at subpixel accuracy. In: *Proceedings of the 8th international conference on computer analysis of images and patterns (CAIP)*. Lecture notes in computer science, vol 1689/1999. Springer, Berlin/New York, pp 534–541
4. Davis CQ, Karul ZZ, Freeman DM (1995) Equivalence of subpixel motion estimators based on optical flow and block matching. In: *Proceedings of the international symposium on computer vision*, Coral Gables, pp 7–12
5. Erdem C, Erdem AT (2001) An illumination invariant algorithm for subpixel accuracy image stabilization and its effect on MPEG-2 video compression. *Signal Process* 16(9): 837–857
6. Fisher RB, Naidu DK (1996) A comparison of algorithms for subpixel peak detection. In: Sanz J (ed) *Advances in image processing, multimedia and machine vision*. Springer-Verlag Berlin/Heidelberg/New York, pp 385–404
7. Foroosh H, Zerubia JB, Berthod M (2002) Extension of phase correlation to subpixel registration. *IEEE Trans Image Process* 11(3):188–200
8. Henkel RD (1998) Fast stereovision with subpixel-precision. In: *Proceedings of the sixth international conference on computer vision*, Bombay, pp 1024–1028
9. Hinz S (2005) Fast and subpixel precise blob detection and attribution. In: *Proceedings of the IEEE international conference on image processing (ICIP)*, vol III, Genoa, pp 457–460
10. Jia H, Yang J, Li X (2010) Minimum variance unbiased subpixel centroid estimation of point image limited by photon shot noise. *J Opt Soc Am* 27(9):2038–2045
11. Karyali IG, Psarakis EZ, Berberidis K, Evangelidis GD (2008) An efficient spatial domain technique for subpixel image registration. *Signal Process* 23(9):711–724
12. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. *Int J Comput Vis* 60(2):91–110
13. MacVicar-Whelan PJ, Binford TO (1981) Intensity discontinuity location to subpixel precision. In: *Proceedings of the international joint conferences on artificial intelligence (IJCAI)*, Vancouver, pp 752–754
14. Morgan GLK, Liu JG, Yan H (2010) Precise subpixel disparity measurement from very narrow baseline stereo. *IEEE Trans Geosci Remote Sens* 48(9):3424–3433
15. Pedersini F, Sarti A, Tubaro S (1997) Estimation and compensation of subpixel edge localization error. *IEEE Trans Pattern Anal Mach Intell* 19(11):1278–1284
16. Stone HS, Orchard MT, Chang E-C, Martucci SA (2001) A fast direct fourier-based algorithm for subpixel registration of images. *IEEE Trans Geosci Remote Sens* 39(10): 2235–2243
17. Suh JW, Jeong J (2004) Fast sub-pixel motion estimation techniques having lower computational complexity. *IEEE Trans Consum Electron* 50(3):968–973
18. Takeshima H, Kaneko T (2008) Image registration using subpixel-shifted images for super-resolution. In: *Proceedings of the 15th IEEE international conference on image processing (ICIP)*, San Diego, pp 2404–2407
19. Tan P, Lin S, Quan L (2008) Subpixel photometric stereo. *IEEE Trans Pattern Anal Mach Intell* 30(8):1460–1471
20. Xiao Y, Fisher RB (2010) Accurate feature extraction and control point correction for camera calibration with a mono-plane target. In: *Proceedings of the international conference on 3D data processing, visualization and transmission (3DPVT)*, Paris, electronic proceedings

Subspace Methods

Kazuhiro Fukui

Department of Computer Science, Graduate School of Systems and Information Engineering, University of Tsukuba, Tsukuba, Japan

Synonyms

Multiple similarity method

Related Concepts

► [Dimensionality Reduction](#); ► [Principal Component Analysis \(PCA\)](#)

Definition

Subspace analysis in computer vision is a generic name to describe a general framework for comparison and classification of subspaces. A typical approach in subspace analysis is the subspace method (SM) that classifies an input pattern vector into several classes based on the minimum distance or angle between the input pattern vector and each class subspace, where a class subspace corresponds to the distribution of pattern vectors of the class in high-dimensional vector space.

Background

Comparison and classification of subspaces has been one of the central problems in computer vision, where an image set of an object to be classified is compactly represented by a subspace in high-dimensional vector space.

The subspace method is one of the most effective classification method in subspace analysis, which was developed by two Japanese researchers, Watanabe and Iijima around 1970, independently [1, 2]. Watanabe and Iijima named their methods the CLAFIC [3] and the multiple similarity method [4], respectively. The concept of the subspace method is derived from the observation that patterns belonging to a class form a compact cluster in high-dimensional vector space, where, for example, a $w \times h$ pixels image pattern is usually represented as a vector in $w \times h$ -dimensional vector space. The compact cluster can be represented by a subspace, which is generated by using Karhunen-Loève (KL) expansion, also known as the principal component analysis (PCA). Note that a subspace is generated for each class, unlike the Eigenface Method [5] in which only one subspace (called eigenspace) is generated.

The SM has been known as one of the most useful methods in pattern recognition field since its algorithm

is very simple and it can handle classification of multiple classes. However, its classification performance was not sufficient for many applications in practice, because class subspaces are generated independently of each other [1]. There is no reason to assume a priori that each class subspace is the optimal linear class subspace in terms of classification performance.

To deal with this problem, the SM has been extended. Two typical extensions are the orthogonal subspace method and the learning subspace methods. The orthogonal subspace method [6] executes the SM to a set of class subspaces that are orthogonalized based on the framework proposed by Fukunaga and Koonitz [7] in learning phase. The orthogonalization is known as a useful operation to boost the performance of angle-based method, such as SM, since class subspaces are usually close to each other in many classification problems.

The learning subspace methods [1, 8, 9] execute the SM to a set of class subspaces, the boundaries between which are adjusted to suppress classification errors for the learning pattern vectors. This adjustment is performed based on the following procedure. First, a learning vector x is classified by using the SM. Then, if x is wrongly classified into an incorrect class subspace L_r , which is not corresponding to the class of x , subspace L_r is slightly rotated into the direction away from x , and in contrast the correct class subspace L_c of x is slightly rotated to the direction close to x . This adjustment is repeated several times for a set of learning vectors until a minimum classification error is achieved.

Moreover, to deal with the nonlinear distribution of pattern vectors, the SM had also been extended to the kernel nonlinear SM [10, 11] by introducing a nonlinear transformation defined by kernel functions.

These extensions aim mainly to improve the classification ability. In addition to such extensions, the generalization of the SM to classification of sets of patterns is also important for many computer vision problems. In order to handle a set of multiple pattern vectors as an input, the SM has been extended to the mutual subspace method (MSM) [12]. The MSM classifies a set of input pattern vectors into several classes based on multiple canonical (principal) angles [13, 14] between the input subspace and class subspaces, where the input subspace is generated from a set of input patterns as class subspaces. The concept of the MSM is closely related to that of the canonical correlation

analysis (CCA) [13]. Actually, the cosine of the i -th smallest canonical angle corresponds to the i -th largest canonical correlation.

The MSM has achieved high performance in recognition of complicated 3D object such as face, using a set of images from image sequence or multi-view images. This success can be mainly explained by the fact that the MSM implicitly utilizes 3D shape information of objects in classification. This is because the similarity between two distributions of various view images of objects reflects the 3D shape similarity between the two objects. To boost the performance of the MSM, it has been further extended to the constrained mutual subspace method (CMSM) [15] and the whitening (or orthogonal) mutual subspace method (WSM) [16], where the relationship among class subspaces is modified to approach orthogonalization in the learning phase. These extensions have boosted the classification ability of the MSM. The MSM and its extensions also have been further extended to the kernel nonlinear SM [17–20].

Theory

Subspace Method

Assume an input vector p and k class subspaces in f -dimensional vector space. The similarity S of the pattern vector p to the i -th class is defined based on either of the length of the projected input vector \hat{p} on the i -th reference subspace [3] or the minimum angle [4] between the input vector p and the i -th class subspace as shown in Fig. 1a. The length of an input vector p is often normalized to 1.0. In this case, these two criteria coincide. In the following explanation, therefore, the angle-based similarity S defined by the following equation will be used:

$$S = \cos^2 \theta = \sum_{i=1}^{d_q} \frac{(p \cdot \phi_i)^2}{\|p\|^2}, \quad (1)$$

where d_q is the dimension of the class subspace and ϕ_i is the i -th f -dimensional orthogonal normal basis vector of the class subspace, which are obtained from applying the principal component analysis (PCA) to a set of patterns of the class. Concretely, these orthonormal basis vectors can be obtained as the eigenvectors of the correlation matrix $\sum_{i=1}^l x_i x_i^\top$ calculated from the l learning patterns $\{x\}$ of the class.

Process Flow of the SM

The whole process of the SM consists of a learning phase and a recognition phase.

In the Learning Phase All k class d_q -dimensional subspaces are generated from a set of pattern vectors of each class by using PCA.

In the Recognition Phase The similarities S of an input vector p to all the k class subspaces are calculated by using Eq. (1). Then, the input vector is classified into the class of the class subspace with highest similarity. If the highest similarity is lower than a threshold value fixed in advance, the input vector is classified into a reject class.

Mutual Subspace Method

Assume an input subspace and class subspaces in f -dimensional vector space. The similarity of the input subspace to the i -th class subspace is defined based on a minimum canonical angle θ_1 [13, 14] between the input subspace and the class subspace, as shown in Fig. 1b.

Given a d_p -dimensional subspace \mathcal{P} and a d_q -dimensional subspace \mathcal{Q} (for convenience, $d_p \leq d_q$) in the f -dimensional vector space, the canonical angles $\{0 \leq \theta_1, \dots, \theta_{d_p} \leq \frac{\pi}{2}\}$ between \mathcal{P} and \mathcal{Q} are uniquely defined as [14]

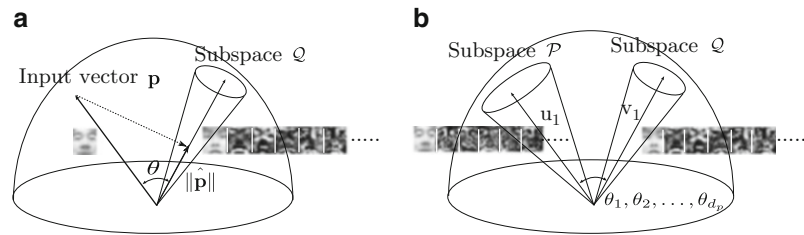
$$\cos^2 \theta_i = \max_{\substack{u_i \perp u_j (i \neq j, i, j = 1 \sim d_p) \\ v_i \perp v_j (i \neq j, i, j = 1 \sim d_q)}} \frac{(u_i \cdot v_i)^2}{\|u_i\|^2 \|v_i\|^2}, \quad (2)$$

where $u_i \in \mathcal{P}, v_i \in \mathcal{Q}, \|u_i\| \neq 0, \|v_i\| \neq 0, (\cdot)$ and $\|$ represent an inner product and a norm, respectively.

Let Φ_i and Ψ_i denote the i -th f -dimensional orthonormal basis vectors of the subspaces \mathcal{P} and \mathcal{Q} , respectively. A practical method of finding the canonical angles is by computing the matrix $X = A^\top B$, where $A = [\Phi_1, \dots, \Phi_{d_p}]$ and $B = [\Psi_1, \dots, \Psi_{d_q}]$. Let $\{\kappa_1, \dots, \kappa_{d_p}\}$ ($\kappa_1 \geq \dots \geq \kappa_{d_p}$) be the singular values of the matrix X . The cosines of canonical angles $\{\theta_1, \dots, \theta_{d_p}\}$ can be obtained as $\{\kappa_1, \dots, \kappa_{d_p}\}$. The original MSM uses only a minimum canonical angle θ_1 to define the similarity. However, since the remaining canonical angles also have information for classification, the value, $\tilde{S} = \frac{1}{d_p} \sum_{i=1}^{d_p} \cos^2 \theta_i$, defined from the

Subspace Methods, Fig. 1

Conceptual illustrations of SM and MSM. (a) Subspace method (SM). (b) Mutual subspace method (MSM)



smallest t canonical angles is often used as the similarity in many computer vision problems. The value \tilde{S} reflects the structural similarity between two subspaces. The whole process of the MSM is the same as that of the SM except that an input vector is replaced by an input subspace.

Application

The subspace methods and their extensions have been applied to various problems [1, 10, 11] of computer vision due to their high general versatility and low computational cost. In particular, the extended SMs have produced remarkable results in optical character recognition (OCR), such as handwriting Chinese character recognition [2, 4], in Japanese industry.

The mutual subspace method has also been demonstrated to be extremely effective for 3D object recognition. In particular, the MSM has been known to be suitable for face recognition [15, 16, 21] because the subspace (called “illumination subspace”), which includes any face image patterns under all possible illumination conditions, can be generated from face images under more than three different illumination conditions [22]. The nonlinear extensions of the MSM have been shown to be further effective for 3D object recognition using image sequences, multi-view images [17–20].

References

- Oja E (1983) Subspace methods of pattern recognition. Research Studies Press, Letchworth
- Kurosawa Y (2007) The engineer's guide to the subspace method. In: ACCV 2007 workshop Subspace 2007, Tokyo, pp 1–8
- Watanabe S, Lambert PF, Kulikowski CA, Buxton JL, Walker R (1967) Evaluation and selection of variables in pattern recognition. In: Tou J (ed) Computer and information sciences. Academic, New York
- Iijima T, Genchi H, Mori K (1973) A theory of character recognition by pattern matching method. In: Proceedings of 1st international conference on pattern recognition (ICPR), pp 50–56
- Turk M, Pentland A (1991) Eigenfaces for recognition. J Cogn Neurosci 3:71–86
- Kittler J (1978) The subspace approach to pattern recognition. Prog Cybern Syst Res 3:92
- Fukunaga K, Koontz W (1970) Application of the Karhunen-Loève expansion to feature selection and ordering. IEEE Trans Comp 19(4):311–318
- Kohonen T, Nemeth G, Jalanko M, Riittinen H (1979) Spectral classification of phonemes by learning subspace methods. In: Proceedings of IEEE international conference on acoustics, speech, and signal processing (ICASSP1979), Washington, DC, vol 4, pp 97–100
- Oja E, Kuusela M (1983) The ALSM algorithm – an improved subspace method of classification. Pattern recognition 16:421–427
- Maeda E, Murase H (1999) Multi-category classification by kernel based nonlinear subspace method. In: Proceedings of IEEE international conference on acoustics, speech, and signal processing (ICASSP1999), Phoenix, vol 2, pp 1025–1028
- Tsuda K (1999) Subspace classifier in the hilbert space. Pattern Recogn Lett 20:513–519
- Maeda K, Watanabe S (1985) A pattern matching method with local structure. Trans IEICE J68-D:345–352 (in Japanese)
- Hotelling H (1936) Relations between two sets of variates. Biometrika 28:321–377
- Chatelin F (1993) Eigenvalues of matrices (enlarged translation of the French publication with masson). Wiley, Chichester
- Fukui K, Yamaguchi O (2003) Face recognition using multi-viewpoint patterns for robot vision. In: 11th international symposium of robotics research (ISRR2003), Siena, pp 192–201
- Kawahara T, Nishiyama M, Kozakaya T, Yamaguchi O (2007) Face recognition based on whitening transformation of distribution of subspaces. ACCV 2007 workshops Subspace2007, Tokyo, pp 97–103
- Sakano H, Mukawa N (2000) Kernel mutual subspace method for robust facial image recognition. In: Fourth international conference on knowledge-based intelligent engineering systems & allied technologies (KES2000), Brighton, vol 1, pp 245–248
- Wolf L, Shashua A (2003) Learning over sets using kernel principal angles. J Mach Learn Res 4: 913–931

19. Fukui K, Stenger B, Yamaguchi O (2006) A framework for 3D object recognition using the kernel constrained mutual subspace method. In: Proceedings of Asian conference on computer vision (ACCV2006), Hyderabad, pp 315–324
20. Fukui K, Stenger B, Yamaguchi O (2007) The kernel orthogonal mutual subspace method and its application to 3D object recognition. In: Proceedings of Asian conference on computer vision (ACCV2007), Tokyo, pp 467–476
21. Yamaguchi O, Fukui K, Maeda K (1998) Face recognition using temporal image sequence. In: Proceedings of IEEE international conference on automatic face and gesture recognition (FG), Nara, pp 318–323
22. Georgiades AS, Belhumeur PN, Kriegman DJ (2001) From few to many: illumination cone models for face recognition under variable lighting and pose. *IEEE Trans Pattern Anal Mach Intell* 23:643–660

Superresolution

► [Subpixel Estimation](#)

Surface Corrugations

► [Surface Roughness](#)

Surface Orientation Histogram (Discrete Version of EGI)

► [Extended Gaussian Image \(EGI\)](#)

Surface Reconstruction

► [Three Dimensional View Integration](#)

Surface Roughness

S. C. Pont
Industrial Design Engineering, Delft University of
Technology, Delft, The Netherlands

Synonyms

[Micro scale structure](#); [Surface corrugations](#); [Surface undulations](#)

Related Concepts

► [Bidirectional Texture Function and 3D Texture](#)

Definition

Surface roughness is structure on the microscale of object surfaces. The illumination of such rough surfaces causes shading, shadowing, interreflections, and occlusion effects on the microscale, resulting in 3D texture, which depends strongly on the viewing direction and on the illumination conditions.

Background

Most natural surfaces are rough on the microscale. This microscale structure can be described mathematically with exact geometrical models, with statistical surface height or attitude (slope) distributions.

Surface roughness causes 3D texture in images, which varies over objects as a function of the local viewing angle and illumination conditions. They can be described by the bidirectional texture function or BTF [3]. Thus, textures of rough objects cannot be texture-mapped, in contradistinction to flat, wallpaper-type textures. They need to be synthesized using surface models or photographed BTFs [6]. Even unresolved 3D texture in an image affects material appearance, through effects on the bidirectional reflectance distribution function (BRDF [7]).

Theory

Since the optical effects due to surface roughness are quite complicated, it is hard to formally derive models for 3D textures. Physics-based optical models can be of a geometrical optical or a statistical nature. Physically exact geometrical models are scarce, because for most surface roughness structures the shadowing and interreflections calculations are intractable. As a summary description of surface roughness, one may use such measures as the probability density of heights, the autocorrelation function of heights, and the probability density of orientations of local microfacets. For example, “bump mapping” techniques [4] in computer

graphics regard only the distribution of orientations and ignore differences in height. Indeed much of the image structure generated by 3D texture is due to the fact that surface microfacets differ in orientation and thus receive different illuminances according to Lambert's law [5]. The bidirectional texture contrast function (BTCF [9]) provides robust guesstimates of the attitude distribution. However, the height distribution is also important because it causes such important effects as vignetting, shadowing and occlusion on the microscale.

The field of surface metrology is concerned with surface roughness measurements and descriptions. Using profilometers, surface profiles of real surfaces can be measured, from which roughness parameters can be derived. Roughness parameters usually are statistical measures over lines or areas of the height profile. Many different parameters are in use and can easily be found in engineering literature and via the Internet [11]. Photometric surface metrology from single images suffers from the bas-relief ambiguity [1] and is in computer vision usually referred to as texture analysis [6].

Open Problems

3D texture provides information which is additional to shading; shading is often primarily dependent on the normal component of the illumination, while 3D texture is primarily dependent on the tangential component of the illumination. The spatial structure of 3D texture gradients, e.g., the "illuminance flow," allows inferences about shape and illumination. Formal solutions to the question how exactly shading and 3D texture combine and interact might further the field of shape from shading.

Experimental Results

Several databases of images of rough surfaces can be found on the Internet [2, 8, 10].

References

1. Belhumeur PN, Kriegman DJ, Yuille AL (1999) The bas-relief ambiguity. *Int J Comput Vis* 35(1):1573–1405

2. CURET, Columbia–Utrecht reflectance and texture database. <http://www.cs.columbia.edu/CAVE/curet>
3. Dana KJ, van Ginneken B (1977) Reflectance and texture of real-world surfaces. In: *Proceedings IEEE computer science conference on computer vision and pattern recognition (CVPR)*
4. Foley JD, van Dam A, Feiner SK, Hughes JF (1990) *Computer graphics, principles and practice*. Addison–Wesley, Reading
5. Lambert JH (1760) *Photometria Sive de Mensure de Gradibus Luminis, Colorum et Umbræ*. Eberhard Klett, Augsburg
6. Mirmehdi M, Xie X, Suri J (2008) *Handbook of texture analysis*. Imperial College Press, London
7. Nicodemus FE, Richmond JC, Hsia, JJ (1977) *Geometrical considerations and nomenclature for reflectance*. National bureau of standards U.S. monograph, vol 160. U.S. Department of Commerce, Washington
8. PhoTex, Photometric 3D texture database. <http://www.taurusstudio.net/research/pmtexdb/index.htm>
9. Pont SC, Koenderink JJ (2005) Bidirectional texture contrast function. *Int J Comput Vis* 67(1/2):17–34
10. Texture Lab PhoTex, Photometric 3D texture database. <http://www.macs.hw.ac.uk/texturelab/resources/databases/Photex/thumbnails.htm>
11. Wikipedia, Roughness. <http://en.wikipedia.org/wiki/Roughness>

Surface Scattering

- [Asperity Scattering](#)

Surface Undulations

- [Surface Roughness](#)

Surfaces

- [Differential Geometry of Surfaces in Three-Dimensional Euclidean Space](#)

Surveillance Camera

- [Pan-Tilt-Zoom \(PTZ\) Camera](#)

SW Cut

- [Swendsen-Wang Cut Algorithm](#)

Swendsen-Wang Algorithm

Adrian Barbu

Department of Statistics, Florida State University,
Tallahassee, FL, USA

Synonyms

Cluster sampling

Related Concepts

► Swendsen-Wang Cut Algorithm

Definition

The Swendsen-Wang (SW) algorithm [1] is an efficient Markov chain Monte Carlo algorithm for sampling from the Ising/Potts model:

$$\pi_{\text{Potts}}(X) = \frac{1}{Z} \exp\left\{ \sum_{\langle i, j \rangle \in E} \beta_{ij} \delta_{X_i = X_j} \right\}, \quad (1)$$

where δ_A is a Boolean function, equal to 1 if condition A is true and 0 otherwise. The Ising/Potts model is defined for a graph $G = \langle V, E \rangle$ and a labeling $X : V \rightarrow \{1, \dots, L\}$. The obtained model is called Ising model [2] when $L = 2$ and Potts model [3] when $L \geq 3$.

Most computer vision applications use $\beta_{ij} = \beta > 0$, also named the ferro-magnetic model, preferring similar colors for neighboring vertices. The Potts models and its extensions are used as prior probabilities in some Bayesian inference tasks.

Background

The SW algorithm was developed in [1] to overcome some of the limitations of the Gibbs sampler [4] in obtaining samples from the Ising/Potts model (1). If one sets $\beta_{ij} = 1/kT$ where T is a parameter called *temperature* and k is a constant, the Gibbs sampler was observed to slow down, obtaining highly correlated

consecutive samples, around a certain temperature named the critical temperature.

In contrast, consecutive samples obtained by the SW algorithm exhibit much smaller correlation at the critical temperature.

Theory

As opposed to the Gibbs sampler [4] that relabels one node at a time, the SW algorithm changes the label of a cluster of nodes in a single move.

The SW algorithm is illustrated in Fig. 1. At each step, the SW method constructs a new set $F \subset E$ of graph edges, also called the active or “on” edges. This is done by initializing $F = \emptyset$ and adding to F any edge $\langle i, j \rangle \in E$ such that $X_i = X_j$ with probability $1 - e^{-\beta_{ij}}$. A connected component of the new graph $G' = (V, F)$ is selected at random, a new label l is chosen at random among the possible labels $\{1, \dots, L\}$, and all nodes in C are relabeled to l . For more details, see Algorithm 1 below.

Alternatively, the labels of all connected components C of the graph $G' = \langle V, F \rangle$ can be flipped independently.

The SW algorithm is run for many iterations, and after a burn-in period that depends on the number of nodes and the coefficients β_{ij} , the labeling states X will follow the posterior probability Eq. (1).

In a modified version by Wolff [5], one may choose a vertex $v \in V$ and grow a connected component C starting at $C = \{v\}$ and following Bernoulli trials on edges adjacent to C that have not been visited yet. This saves some computation in the clustering step,

Algorithm 1 The Swendsen-Wang Algorithm

Given: Graph $G = \langle V, E \rangle$, $V = (V_1, \dots, V_n)$ and the Ising/Potts model from Eq. (1).

Input: Current labeling state $X = (X_1, \dots, X_n)$.

Output: New labeling state X' .

Set $F = \emptyset$

for all edges $e = \langle i, j \rangle \in E$ with $X_i = X_j$ **do**

Sample $u_{ij} \sim \text{Bernoulli}(p_{ij})$ with $p_{ij} = 1 - e^{-\beta_{ij}}$

if $u_{ij} = 1$ **then**

$F \leftarrow F \cup \{\langle i, j \rangle\}$

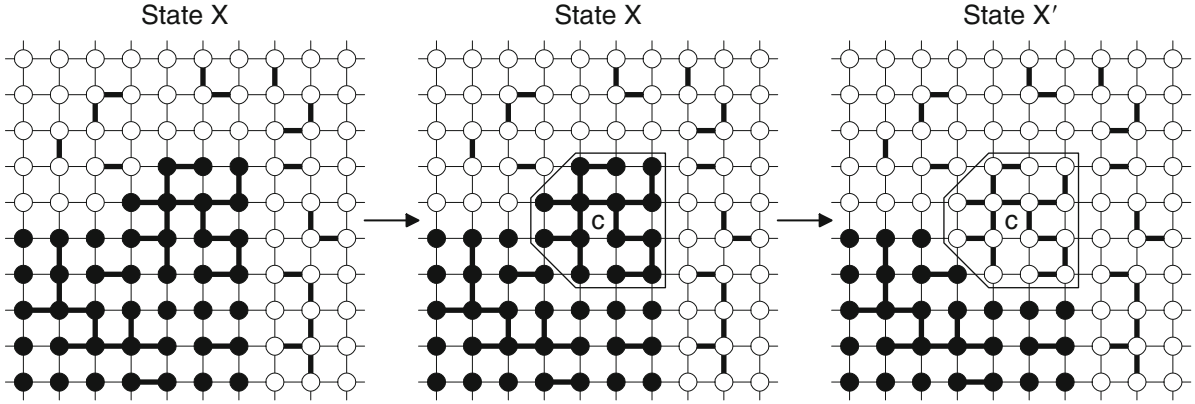
end if

end for

Pick a connected component C of $G' = (V, F)$ at random.

Sample $l \sim \text{Unif}\{1, \dots, L\}$

Set $X'_i = l, \forall i \in C, X'_i = X_i, \forall i \notin C$



Swendsen-Wang Algorithm, Fig. 1 The Swendsen-Wang method. *Left:* In the current labeling state X (*left*), the graph edges (shown with *thin lines*) between same label nodes are turned “on” with probability $p_{ij} = 1 - e^{-\beta_{ij}}$. *Middle:* A

connected component C of the graph of “on” edges is randomly selected, and a new label l is randomly chosen. *Right:* All nodes in C are relabeled with label l , obtaining the new labeling state X'

and bigger components have a higher chance to be selected.

The SW method described above is different from what was presented in the original paper [1]. This description follows the interpretation of Edward and Sokal [6], where the variables $u_{ij} |_{\langle i,j \rangle \in E}$ are collected into the vector U and the set of active edges is $F(U) = \{\langle i,j \rangle \in E, u_{ij} = 1\}$. The SW algorithm is explained as an auxiliary variable method that samples from the joint model

$$p_{ES}(X, U) \propto \prod_{\langle i,j \rangle \in E} [(1 - p_{ij}) \delta_{u_{ij}=0} + p_{ij} \delta_{u_{ij}=1} \cdot \delta_{x_i=x_j}] \\ \propto (1 - p_{ij})^{|E \setminus F(U)|} \cdot p_{ij}^{|F(U)|} \cdot \prod_{\langle i,j \rangle \in F(U)} \delta_{x_i=x_j}.$$

The SW algorithm samples from the above joint model $p_{ES}(X, U)$ by alternatively sampling $p_{ES}(U|X)$ and $p_{ES}(X|U)$. Note that the sampling of $p_{ES}(U|X)$ is exactly the part of the SW algorithm that constructs the random edges F , while the sampling of $p_{ES}(X|U)$ is the part that flips the labels of one or all the connected components C of the graph $G' = \langle V, F \rangle$. By sampling (X, U) from the joint model $p_{ES}(X, U)$, the labelings X will follow the marginal $p_{ES}(X)$ which is exactly the Potts model $p_{Potts}(X)$. On the other hand, the random edges U follow the marginal $p_{ES}(U)$ which is the random cluster model.

Another explanation of the SW algorithm is due to Higdon [7] through the perspective of slice sampling and decoupling. He also introduces partial decoupling, which gives a data-driven clustering step.

The SW algorithm was generalized to arbitrary probabilities in [8], by interpreting it as a Metropolis-Hastings [9, 10] step. This generalization is named the Swendsen-Wang cut algorithm.

An exact sampling method for the Potts model using the SW algorithm was developed by Huber [11], based on coupling from the past [12]. This method eliminates the need for the burn-in period for obtaining samples; however, it is quite conservative and can only be used in practice for small graphs.

Application

Due to being restricted to the Ising/Potts model, there are only a few applications of the SW algorithm to computer vision.

Higdon [7] introduced partial decoupling and presented an application of SW to image reconstruction from positronic emission tomography (PET) data. The SW algorithm with partial decoupling was also used in [13] for texture segmentation using a model with Potts prior and a data term. The SW algorithm was compared with the Gibbs sampler and SW with partial decoupling in [14].

Morris [15] introduced a higher-order prior model named the “chien” model that is not based on pairwise interactions, but on 3×3 cliques. He used the SW algorithm with partial decoupling for obtaining samples from this model.

The SW cut algorithm, a SW generalization to arbitrary probabilities and edge weights, has seen many applications to image, motion, and object segmentation as well as stereo matching and curve grouping, to name only a few. For more details, see the SW cut entry of the encyclopedia.

References

1. Swendsen R, Wang J (1987) Nonuniversal critical dynamics in Monte Carlo simulations. *Phys Rev Lett* 58(2):86–88
2. Ising E (1925) Beitrag zur theorie des ferromagnetismus. *Zeitschrift fur Physik A Hadrons and Nuclei* 31(1):253–258
3. Potts R (1952) Some generalized order-disorder transformations. In: *Mathematical proceedings of the Cambridge philosophical society*, vol 48. Cambridge University Press, Cambridge, pp 106–109
4. Geman S, Geman D, Relaxation S (1984) Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans Pattern Anal Mach Intell* 6(2):721–741
5. Wolff U (1989) Collective Monte Carlo updating for spin systems. *Phys Rev Lett* 62(4):361–364
6. Edwards R, Sokal A (1988) Generalization of the Fortuin-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm. *Phys Rev D* 38(6):2009–2012
7. Higdon D (1998) Auxiliary variable methods for Markov Chain Monte Carlo with applications. *J Am Stat Assoc* 93(442):585–595
8. Barbu A, Zhu S (2005) Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *IEEE Trans Pattern Anal Mach Intell* 27(8):1239–1253
9. Metropolis N, Rosenbluth A, Rosenbluth M, Teller A, Teller E, et al (1953) Equation of state calculations by fast computing machines. *J Chem Phys* 21(6):1087
10. Hastings W (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* 57(1):97
11. Huber M (2003) A bounding chain for Swendsen-Wang. *Random Struct Algorithm* 22(1):43–59
12. Propp J, Wilson D (1996) Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Struct Algorithm* 9(1–2):223–252
13. Barker S, Rayner P (2002) Unsupervised image segmentation. In: *Proceedings of the ICASSP 2002, Orlando*, vol 5. pp 2757–2760
14. Hurn M (1997) Difficulties in the use of auxiliary variables in Markov chain Monte Carlo methods. *Stat Comput* 7(1):35–44
15. Morris R (1999) Auxiliary variables for Markov random fields with higher order interactions. *EMMCVPR, York*, pp 731–731

Swendsen-Wang Cut Algorithm

Adrian Barbu

Department of Statistics, Florida State University,
Tallahassee, FL, USA

Synonyms

SW cut

Related Concepts

► [Simulated Annealing](#); ► [Swendsen-Wang Algorithm](#)

Definition

The Swendsen-Wang Cut algorithm is an efficient Markov chain Monte Carlo (MCMC) algorithm for sampling arbitrary distributions $p(X)$ defined on partitions X of a graph $G = \langle V, E \rangle$. It uses a set of weights on the graph edges to form data-driven clusters and change the label of an entire cluster in one move.

Background

Many computer vision problems can be formulated as the optimization of an energy or probability function defined over the space of partitions or labelings of a graph. Efficient algorithms such as graph cuts [1], belief propagation [2], or dual decomposition [3] can be used for certain types of energy functions.

However, for general forms of the energy or probability functions, many times one has to resort to stochastic relaxation (Gibbs sampler) [4] methods that relabel one node at a time. Such algorithms can be very slow in practice, especially when strong regularization is imposed on the partitions.

For Ising/Potts models [5, 6], the Swendsen-Wang (SW) algorithm [7, 8] was designed to overcome some of the limitations of the stochastic relaxation. The SW algorithm achieves speedup by relabeling a cluster of adjacent nodes of the graph in a single move.

Theory

The Swendsen-Wang cuts algorithm is a generalization of the Swendsen-Wang algorithm to arbitrary distributions by adding an acceptance step for the proposed MCMC cluster relabeling move. The acceptance probability can be computed efficiently using Eq. (3).

Let $G = \langle V, E \rangle$ be a graph, $q_e \in [0, 1], \forall e \in E$ a set of weights on the graph edges, and $p(X)$ a probability defined for any partition or labeling $X = (X_1, \dots, X_n)$ of the graph nodes $V = (V_1, \dots, V_n)$. The probability $p(X)$ can be defined up to a constant. Assume also given a probability mass function $q(l|C, X)$ defined over the set possible labels l , given a labeling state X and a set of nodes $C \subset V$. The function $q(l|C, X)$ could be as simple as a uniform distribution or can be driven by the image data.

For any labeling X of the graph nodes and any label l , define $V_l(X)$ to be the set of nodes with label l :

$$V_l(X) = \{i \in V, X_i = l\}. \quad (1)$$

For any two subsets $C, D \subset V$, define the *cut* from C to D as the set of edges:

$$C(C, D) = \{e = \langle i, j \rangle \in E, i \in C, j \in D\}. \quad (2)$$

With these notations, one move of the SW cut algorithm proceeds as follows, also illustrated in Fig. 1.

The acceptance probability $\alpha(X \rightarrow X')$ has a factor defined in terms of the cuts $C(C, V_l(X) \setminus C)$ and $C(C, V_{l'}(X) \setminus C)$ that contain the edges from C toward the nodes with the old label l and the proposed new label l' , respectively. These cuts are shown with dotted lines in Fig. 1.

The algorithm was observed to be hundreds of times faster than the Gibbs sampler when the edge weights q_e were chosen appropriately. The edge weights should approximate the probability that the two adjacent nodes belong to the same label. This can be achieved using the image information available at the graph nodes and could be learned in a discriminative way, for example, using boosting or regression.

The SW cut algorithm can be used for maximum a posteriori (MAP) estimation using a simulated annealing schedule, in which the probability

$p(X)$ is raised to increasingly larger powers, forcing the sampling to focus on the labelings X of highest probability.

A Wolff [9] version of the SW cut algorithm has also been proposed [10, 11], which grows a connected component from a seed node. This version reduces the amount of computation required for the clustering step, and larger connected components have a higher chance to be selected.

Application

The SWC algorithm has many applications in computer vision. One application was image segmentation [10, 12], where the graph had image superpixels as nodes and edges based on the superpixel adjacency. The edge weights q_e were based on the similarity between the intensity histograms of the corresponding superpixels. The probability $p(X)$ was a Bayesian

Algorithm 1 The Swendsen-Wang Cut Algorithm

Given: Weighted graph $G = \langle V, E \rangle$, $V = (V_1, \dots, V_n)$ and a probability $p(X)$.

Input: Current labeling state $X = (X_1, \dots, X_n)$.

Output: New labeling state X' .

Set $F = \emptyset$

for all edges $e = \langle i, j \rangle \in E$ with $X_i = X_j$ **do**

Sample $u \sim \text{Bernoulli}(q_e)$

if $u = 1$ **then**

$F \leftarrow F \cup \{\langle i, j \rangle\}$

end if

end for

Pick a connected component C of $G' = (V, F)$ at random, with some label $X_C = l$.

Sample a new label $l' \sim q(l'|C, X)$

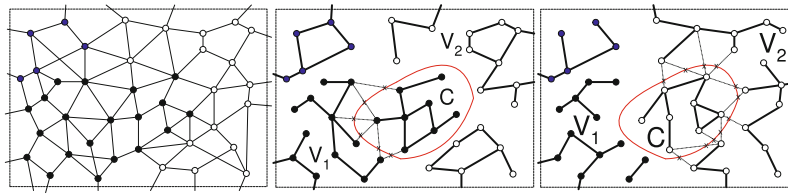
Sample $u \sim \text{Bernoulli}(\alpha(X \rightarrow X'))$ where $X'_i = l', \forall i \in C, X'_i = X_i, \forall i \notin C$ and

$$\alpha(X \rightarrow X') = \min \left(1, \frac{\prod_{e \in C(C, V_{l'}(X) \setminus C)} (1 - q_e)}{\prod_{e \in C(C, V_l(X) \setminus C)} (1 - q_e)} \cdot \frac{q(l|C, X')}{q(l'|C, X)} \cdot \frac{p(X')}{p(X)} \right). \quad (3)$$

if $u = 0$ **then**

Reject the move i.e. make $X' = X$.

end if



Swendsen-Wang Cut Algorithm, Fig. 1 Illustration of the SW cut algorithm. *Left:* A graph $G = \langle V, E \rangle$ with a labeling X . *Middle:* At each step, a new graph $G' = \langle V, F \rangle$ is formed by going through all edges between same label nodes and keeping

an edge e with probability given by its weight q_e . *Right:* Then a connected component C of G' is selected at random, a new label l' is chosen for C , and if the move is accepted, the new labeling state has $X_C = l'$

model with a prior based on connected components and was maximized by SW cut with simulated annealing. Experiments showed that the SW cut algorithm was two orders or magnitude faster in total CPU time when compared to the Gibbs sampler tuned to obtain the same optimal result.

Other applications of the SW cut algorithm include curve grouping [12], dense stereo matching [10, 13], motion segmentation [11, 14] and estimation [15], object segmentation [16], and task allocation [17] in robotics. Furthermore, the SW cut algorithm was used for discovering composite features for object detection [18] and for graph matching [19].

Open Problems

In problems, higher-level objects can Sometimes be detected only when at least three components are considered simultaneously. For such cases, it is an open question how to generalize the SW cut algorithm to use higher-order cliques instead of graph edges to construct the clusters.

For example, finding lines in point clouds can be seen as the problem of labeling the points that belong to each line and the remaining points as background. But since nontrivial colinearity exists only between at least three points, it is difficult to define edge weights directly between the points. A generalization of the SW cut algorithm could use triplets of collinear points instead of graph edges to form point clusters that are likely to be on the same line. It is not known how exactly to form the clusters from such higher-order cliques and what is the acceptance probability for relabeling such a cluster.

References

1. Boykov Y, Veksler O, Zabih R (2002) Fast approximate energy minimization via graph cuts. *IEEE Trans Pattern Anal Mach Intell* 23(11):1222–1239
2. Yedidia J, Freeman W, Weiss Y (2001) Generalized belief propagation. In: *NIPS 2001, Vancouver*, pp 689–695
3. Komodakis N, Paragios N, Tziritas G (2007) MRF optimization via dual decomposition: message-passing revisited. In: *ICCV 2007, Rio de Janeiro*
4. Geman S, Geman D, Relaxation S (1984) Gibbs distributions, and the bayesian restoration of images. *IEEE Trans Pattern Anal Mach Intell* 6(2):721–741
5. Ising E (1925) Beitrag zur theorie des ferromagnetismus. *Zeitschrift fur Physik A Hadrons and Nuclei* 31(1):253–258
6. Potts R (1952) Some generalized order-disorder transformations. In: *Mathematical proceedings of the cambridge philosophical society*, vol 48. Cambridge University Press, Cambridge, pp 106–109
7. Swendsen R, Wang J (1987) Nonuniversal critical dynamics in Monte Carlo simulations. *Phys Rev Lett* 58(2):86–88
8. Edwards R, Sokal A (1988) Generalization of the fortuin-kasteleyn-swendsen-wang representation and monte carlo algorithm. *Phys Rev D* 38(6):2009–2012
9. Wolff U (1989) Collective Monte Carlo updating for spin systems. *Phys Rev Lett* 62(4):361–364
10. Barbu A, Zhu S (2005) Generalizing Swendsen-Wang to sampling arbitrary posterior probabilities. *IEEE Trans Pattern Anal Mach Intell* 27(8):1239–1253
11. Barbu A, Zhu S (2007) Generalizing swendsen-wang for image analysis. *J Comput Graph Stat* 16(4):877–900
12. Barbu A, Zhu S (2003) Graph Partition by Swendsen-Wang Cuts. In: *ICCV 2003, Nice*, p 320
13. Kim W, Park J, Lee K (2009) Stereo matching using population-based MCMC. *Int J Comput Vis* 83(2): 195–209
14. Barbu A, Zhu S (2004) Multigrid and multi-level Swendsen-Wang cuts for hierarchic graph partition. In: *CVPR 2004, Washington, DC*
15. Barbu A, Yuille A (2004) Motion estimation by Swendsen-Wang cuts. In: *CVPR 2004, Washington, DC*
16. Wang J, Betke M, Gu E (2005) MosaicShape: stochastic region grouping with shape prior. In: *IEEE conference on computer vision pattern recognition (CVPR 2005), San Diego*, vol 1. pp 902–908

17. Zhang K, Collins E, Barbu A (2010) A novel stochastic clustering auction for task allocation in multi-robot teams. In: IROS 2010, Taipei
18. Han F, Shan Y, Sawhney H, Kumar R (2008) Discovering class specific composite features through discriminative sampling with swendsen-wang cut. In: IEEE conference on computer vision pattern recognition (CVPR 2008), Anchorage, pp 1–8
19. Lin L, Zhu S, Wang Y (2007) Layered graph match with graph editing. In: IEEE conference on computer vision pattern recognition (CVPR 2007), Minneapolis, pp 1–8

Symmetry Detection

► [Computational Symmetry](#)

Symmetry-Based X

► [Computational Symmetry](#)