ICP

▸Iterative Closest Point (ICP)

Illumination Estimation, Illuminant Estimation

Stephen Lin
Microsoft Research Asia, Beijing Sigma Center,
Beijing, China

Related Concepts

▸Color Constancy; ▸Incident Light Measurement

Definition

The purpose of illumination estimation is to determine the direction, intensity, and/or color of the lighting in a scene. In contrast to direct measurement of lighting, the illumination information is inferred from cues within the scene, without use of a special probe or color calibration chart.

Background

The appearance of objects and scenes can vary considerably with respect to illumination conditions. In [1], differences in face appearance due to lighting were found to be greater than those due to identity. Since such appearance variations can affect the performance of certain computer vision algorithms, much research has focused on illumination estimation, so that lighting can be accounted for in image understanding.

To simplify inference, methods for illumination estimation typically assume that the illumination originates from distant light sources. With this assumption, the illumination can be considered to be uniform across the scene, such that only a single lighting condition needs to be estimated. Most techniques perform this estimation on a single input image, as this allows for wider applicability.

Methods

Different image cues have been utilized to estimate illumination. Several techniques categorized by cue are described in the following.

Shading

Many methods for illumination estimation are based on an analysis of shading over the surface of an object. They typically utilize the relationship between shading and lighting described by the Lambertian reflectance model:

$$I(x) = \rho(x) N(x) \cdot L$$

where x indexes the shaded image pixels, I denotes image intensity (shading), ρ is the albedo, $N(x)$ is the surface normal, and L is the light vector that encodes the direction and magnitude of illumination. To solve for L, shading-based techniques for illumination estimation generally assume the surface of interest to have a uniform albedo and a known geometry. If the absolute albedo value is unknown, then L can be estimated up to an unknown scale factor.

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

While some methods focus on recovering only the direction of a single illuminant [2, 3], most address the more common scenario of multiple illumination sources. Hougen and Ahuja [4] solve a set of linear equations to determine light intensities from a set of sampled directions. Yang and Yuille [5] use image intensities and known surface normals at occluding boundaries to constrain illuminant directions. Ramamoorthi and Hanrahan [6] compute a low-frequency illumination distribution from a deconvolution of reflectance and lighting. Zhang and Yang [7] estimate lighting directions from critical points which have surface normals perpendicular to an illuminant direction. Based on this, Wang and Samaras [8] segmented images into regions of uniform lighting and then performed estimation by recursive least-squares fitting of the Lambertian reflectance model to these regions.

Illumination may alternatively be estimated by uncalibrated photometric stereo [9], without the need for known albedos and surface normals. This approach requires a set of images taken under different lighting conditions as input.

Cast Shadows

Several techniques analyze cast shadows for illumination estimation. For an object of known shape, the shadows that it casts provide constraints on the lighting directions and their corresponding intensities. Sato et al. [10–13] formulated these constraints as a system of equations in terms of observed brightness values within shadows and a set of sampled lighting directions at which source intensities are to be solved. These methods require a single input image for objects that cast shadows onto a uniform-colored surface; two images are needed to cancel out the effects of color variation for surfaces with texture. This approach was extended by Okabe et al. [14] to a lighting representation of Haar wavelets. Kim and Hong [15] later proposed a single-image method that handles surface texture by incorporating regularization and some user-specified information.

Specular Reflections

Some methods consider specular reflections in estimating illumination. From the locations of specular reflections on an object of known shape, these techniques compute the corresponding light source directions according to the mirror reflection property. This approach was used by Nishino et al. [16] to obtain an initial approximation of the illumination distribution, which is then refined using a more sophisticated model of reflectance. Illumination cues from specular reflections are combined with those from shading and shadows by Li et al. [17] to minimize the effects of scene texture on lighting estimation. Without needing explicit object shape recovery, Nishino et al. [18] and Wang et al. [19] proposed to estimate lighting from specular reflections on human eyes, which are highly reflective and have a similar shape from person to person.

Color

Much research focuses on estimating the color of illumination, rather than the directional distribution. Early methods solve for light color based on assumed properties of the imaged scene, such as the average color being achromatic [20] or that the scene contains a maximally reflective white patch [21]. More recent techniques are guided by more detailed knowledge about illumination and surface colors in the natural world. A statistical model of lights and surfaces from training data is used by Brainard and Freeman [22] and Finlayson et al. [23] to obtain a solution. Knowledge about illuminations and surfaces is instead used by Forsyth [24] and Finlayson et al. [25] to constrain the range of illuminant colors that could possibly result in the observed image. A comprehensive review of illumination color estimation methods is provided in [26].

Application

Illumination estimation has been employed in various applications based on appearance modeling. In [18], lighting estimates from eye reflections are used for robust face recognition under varying illumination conditions. Estimates of scene illumination have also been used to realistically composite virtual objects into an image in an illumination-consistent manner [27, 28]. In digital cameras, methods for estimating illumination color are incorporated into automatic white balance algorithms. Recent methods for estimating light color, however, have been found to be inadequate in improving color-based object recognition [29].

References

1. Moses Y, Adini Y, Ullman S (1994) Face recognition: the problem of compensating for changes in illumination direction. In: Proceedings of European conference on computer vision (ECCV). Springer, Heidelberg/Berlin, pp 286–296
2. Zheng Q, Chellappa R (1991) Estimation of illuminant direction, albedo, and shape from shading. IEEE Trans Pattern Anal Mach Intell 13:680–702
3. Samaras D, Metaxas D (1999) Coupled lighting direction and shape estimation from single images. In: Proceedings of the international conference on computer vision. IEEE Computer Society, Washington, DC, pp 868–874
4. Hougen DR, Ahuja N (1993) Estimation of the light source distribution and its use in integrated shape recovery from stereo and shading. In: Proceedings of the international conference on computer vision. IEEE Computer Society, Washington, DC, pp 148–155
5. Yang Y, Yuille AL (1991) Sources from shading. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE Computer Society, Washington, DC, pp 534–539
6. Ramamoorthi R, Hanrahan P (2001) A signal-processing framework for inverse rendering. In: Proceedings of ACM SIGGRAPH. ACM, New York, pp 117–128
7. Zhang Y, Yang YH (2001) Multiple illuminant direction detection with application to image synthesis. IEEE Trans Pattern Anal Mach Intell 23:915–920
8. Wang Y, Samaras D (2002) Estimation of multiple illuminants from a single image of arbitrary known geometry. In: Proceedings of European conference on computer vision (ECCV). Lecture notes in computer science, vol 2352. Springer, Berlin/Heidelberg, pp 272–288
9. Basri R, Jacobs D, Kemelmacher I (2007) Photometric stereo with general, unknown lighting. Int J Comput Vis 72:239–257
10. Sato I, Sato Y, Ikeuchi K (1999) Illumination distribution from brightness in shadows: adaptive estimation of illumination distribution with unknown reflectance properties in shadow regions. In: Proceedings of the international conference on computer vision. IEEE Computer Society, Washington, DC, pp 875–883
11. Sato I, Sato Y, Ikeuchi K (1999) Illumination distribution from shadows. In: Proceeding of the IEEE Conference on computer vision and pattern recognition (CVPR). IEEE Computer Society, Washington, DC, pp 306–312
12. Sato I, Sato Y, Ikeuchi K (2001) Stability issues in recovering illumination distribution from brightness in shadows. Proc IEEE Conf Comput Vis Pattern Recognit (CVPR) II:400–407
13. Sato I, Sato Y, Ikeuchi K (2003) Illumination from shadows. IEEE Trans Pattern Anal Mach Intell 25:290–300
14. Okabe T, Sato I, Sato Y (2004) Spherical harmonics vs. haar wavelets: basis for recovering illumination from cast shadows. Proc IEEE Conf Comput Vis Pattern Recognit (CVPR) I:50–57
15. Kim T, Hong K (2005) A practical single image based approach for estimating illumination distribution from shadows. In: Proceedings of the international conference on computer vision. IEEE Computer Society, Washington, DC, pp 266–271
16. Nishino K, Zhang Z, Ikeuchi K (2001) Determining reflectance parameters and illumination distribution from sparse set of images for viewdependent image synthesis. In: Proceedings of the international conference on computer vision. IEEE Computer Society, Washington, DC, pp 599–606
17. Li Y, Lin S, Lu H, Shum HY (2003) Multiple-cue illumination estimation in textured scenes. In: Proceedings of the international conference on computer vision. IEEE Computer Society, Washington, DC, pp 1366–1373
18. Nishino K, Belhumeur P, Nayar S (2005) Using eye reflections for face recognition under varying illumination. Proc Int Conf Comput Vis I:519–526
19. Wang H, Lin S, Liu X, Kang SB (2005) Separating reflections in human iris images for illumination estimation. In: Proceedings of the international conference on computer vision. IEEE Computer Society, Washington, DC, pp 1691–1698
20. Buchsbaum G (1980) A spatial processor model for object colour perception. J Franklin Inst 310:1–26
21. Land EH (1977) The retinex theory of color vision. Sci Am 237:108–128
22. Brainard DH, Freeman WT (1997) Bayesian color constancy. J Opt Soc Am A 14:1393–1411
23. Finlayson GD, Hordley S, Hubel PM (2001) Color by correlation: a simple, unifying framework for color constancy. IEEE Trans Pattern Anal Mach Intell 23:1209–1221
24. Forsyth DA (1990) A novel algorithm for colour constancy. Int J Comput Vis 5:5–36
25. Finlayson GD, Hordley S, Tastl I (2006) Gamut constrained illuminant estimation. Int J Comput Vis 67:93–109
26. Ebner M (2007) Color constancy. Wiley, Chichester
27. Sato I, Sato Y, Ikeuchi K (1999) Acquiring a radiance distribution to superimpose virtual objects onto a real scene. IEEE Trans Vis Comput Graph 5:1–12
28. Lalonde JF, Efros AA, Narasimhan SG (2009) Estimating natural illumination from a single outdoor image. In: Proceedings of the international conference on computer Vision. IEEE Computer Society, Washington, DC
29. Funt B, Barnard K, Martin L (1998) Is colour constancy good enough? In: Proceedings of the European conference on computer vision (ECCV). Springer, London, pp 445–459

Image Alignment

▸Image Registration

Image Decompositions

Marshall F. Tappen
University of Central Florida, Orlando, FL, USA

Definition

An image decomposition is the result of a mathematical transformation of an image into a new set of images that represent different aspects of the input image or scene pictured in that image. The original image can typically be reconstructed from these new images.

Background

While images are primarily stored as an array of pixel values, an image can be represented in a number of different ways. For instance, an image can be easily transformed into two images, one containing the high-frequency variation in the input image and a second containing the low-frequency variation. This process decomposes the input image into two images, each of which expresses different information about the original image.

This process is useful when further processing will treat these two images differently. If the decomposition is chosen correctly, the image is decomposed into a set of images that can each be processed uniformly. Thus, the decomposition facilitates adaptive processing of the content of an image.

Theory

Image decompositions can be roughly divided into two different types of decompositions, image-based decompositions and intrinsic image decompositions. Image-based decompositions represent the image itself using new images, while the intrinsic image decompositions reflect the content of the scene pictured in the image itself.

Image-Based Decompositions

Similar to the background example above, many image-based decompositions focus on representing multi-scale frequency content in the scene. The Gaussian pyramid is one of the most basic decompositions representing multi-scale content. The decomposition consists of a set of images of progressively smaller resolution, with each image being one level of the pyramid. Each level is created by filtering the image at the level below, then downsampling the result. This creates a multi-resolution set of images.

Depending on the application, the usefulness of the Gaussian pyramid may be limited because each level contains redundant information. This can be eliminated by modifying the pyramid creation process to create a Laplacian pyramid [1]. In the Laplacian pyramid, the input image is progressively downsampled. The image at level i in the Laplacian pyramid is computed by taking the difference between the ith level of the Gaussian pyramid and the upsampled version of level $i + 1$, which has been downsampled from the ith level of the Gaussian pyramid. Effectively, each level of the Laplacian pyramid expresses the image information at a particular scale. Figure 1 shows an example of the Laplacian decomposition of an image.

In [2], Simoncelli et al. extended this decomposition process to also separate orientation into different images, creating the steerable pyramid decomposition. Similar decompositions can also be generated by using a different process to separate the images. In [3], the bilateral filter is used to generate a two-image decomposition.

These decompositions are also connected to other image transformations, particularly wavelets. The connections are discussed in [2].

Intrinsic Image Decompositions

While image-based decompositions are focused on the pixel values themselves, intrinsic image decompositions create images that are based on the content of the scene. The intrinsic image decomposition is based on the intrinsic image approach for representing scene characteristics. In this approach, each intrinsic characteristic of the scene is represented by a distinct image. In the intrinsic image decomposition, these images are chosen to both represent intrinsic characteristics and image content.

In [4], Weiss uses video data to separate an image into illumination, or shading, and albedo components. In this decomposition, an input image pixel at location n, $I(n)$, is equal to the product of a shading image and an albedo image, or $I(n) = S(n) \times A(n)$. In [5] and [6], Tappen et al. show how the intrinsic image decomposition can be computed from a single image. Figure 2 shows an example of an intrinsic image decomposition for the image on the left.

Application

Image decompositions are frequently used to generate images that processed separately. In [7], Portilla et al. use the steerable pyramid to denoise images. Heeger and Bergen showed that texture can be generated by

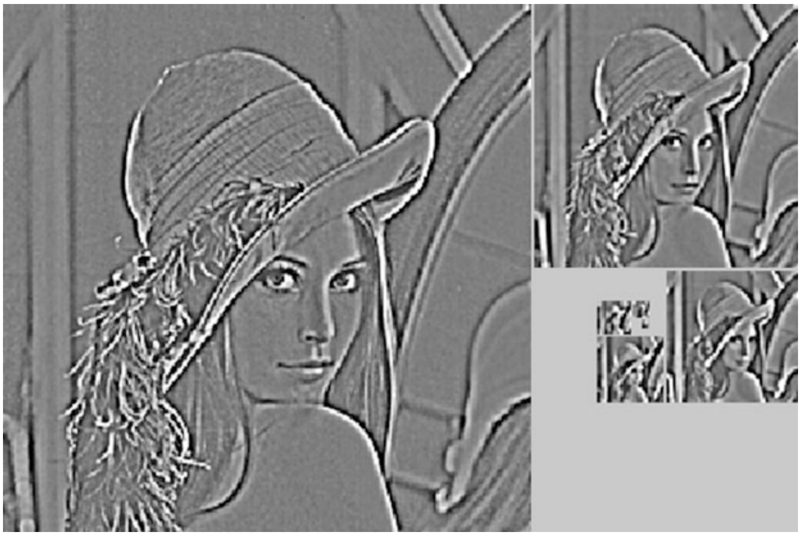


Image Decompositions, Fig. 1 These images are a Laplacian pyramid created from the well-known *Lena* image. Each image captures the variation at a specific scale

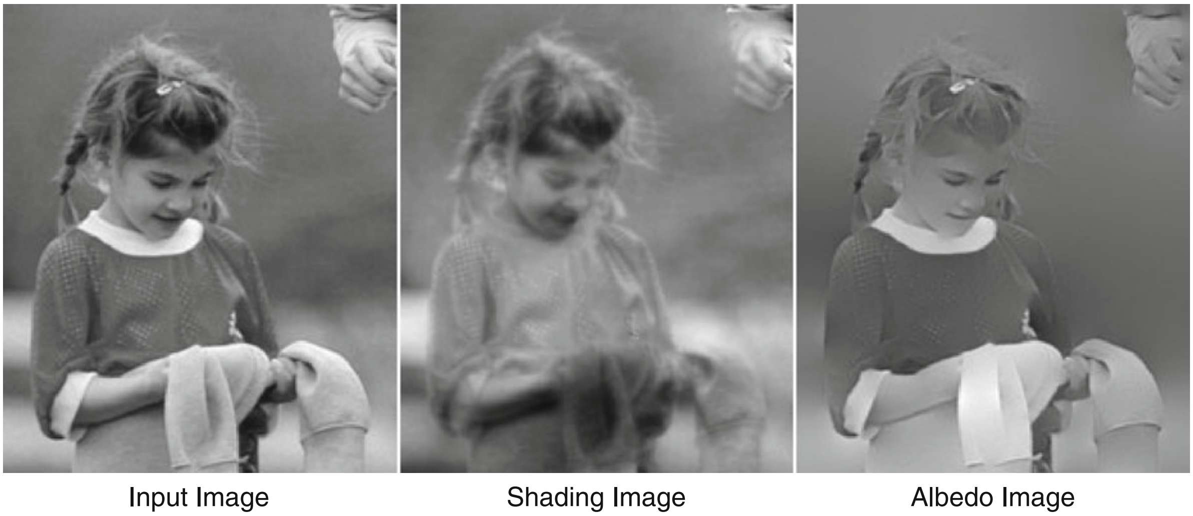


Image Decompositions, Fig. 2 An example of an intrinsic image decomposition. The image on the *left* is decomposed into shading and albedo components

forcing the marginal histograms of the levels of a steerable pyramid to match those of a pyramid generated from a reference image [8]. More complete measures of statistical similarity are used in [9], leading to improved synthesis results.

As mentioned earlier, the bilateral filter is used in [3] to separate the image into large- and fine-scale variations to combine images taken under different illumination. In [10], Bousseau et al. describe how user input can improve intrinsic image decompositions and demonstrate how they can be applied for graphics applications.

References

1. Burt P, Adelson E (1983) The Laplacian pyramid as a compact image code. IEEE Trans Commun 31(4):532–540
2. Simoncelli EP, Freeman WT, Adelson EH, Heeger DJ (1992) Shiftable multiscale transforms. IEEE Trans Inf Theory 38(2):587–607
3. Eisemann E, Durand F (2004) Flash photography enhancement via intrinsic relighting. ACM Trans Graph 23:673–678
4. Weiss Y (2001) Deriving intrinsic images from image sequences. In: IEEE international conference on computer vision, Vancouver, vol 2, p 68
5. Tappen MF, Freeman WT, Adelson EH (2005) Recovering intrinsic images from a single image. IEEE Trans Pattern Anal Mach Intell 27(9):1459–1472

6. Tappen MF, Adelson EH, Freeman WT (2006) Estimating intrinsic component images using non-linear regression. In: The proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR), New York, vol 2, pp 1992–1999
7. Portilla J, Strela V, Wainwright M, Simoncelli E (2003) Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Trans Image Process 12(11): 1338–1351
8. Heeger DJ, Bergen JR (1995) Pyramid-based texture analysis/synthesis. In: Proceedings of ACM SIGGRAPH 1995. ACM, New York, pp 229–238
9. Portilla J, Simoncelli EP (2000) A parametric texture model based on joint statistics of complex wavelet coefficients. Int J Comput Vis 40(1):49–70
10. Bousseau A, Paris S, Durand F (2009) User assisted intrinsic images. ACM Trans Graph (Proceedings of SIGGRAPH Asia 2009) 28(5)

Image Enhancement and Restoration

Guoshen Yu[1] and Guillermo Sapiro[2]
[1]Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA
[2]Electrical and Computer Engineering, Computer Science, and Biomedical Engineering, Duke University, Durham, NC, USA

Synonyms

Image inverse problems

Related Concepts

►Denoising; ►Image-Based Modeling; ►Inpainting

Definition

Image enhancement and restoration is a procedure that attempts to improve the image quality by removing the degradation while preserving the underlying image characteristics.

Background

Image quality is often deteriorated during acquisition, compression, and transmission. Typical degradations include image blur introduced by lens out-of-focus, resolution downgrade due to acquisition equipment pixel limitation, noise spots introduced at high ISO, and JPEG block artifact, as illustrated in Fig. 1. Image enhancement and restoration is a procedure that attempts to improve the image quality by removing the degradation while preserving the underlying image characteristics. For some specific degradations as mentioned above, image enhancement and restoration is also known as deblurring, super-resolution zooming, denoising, and deblocking. While jointly addressed here and in most of the literature, *restoration* often refers to the case where one attempts to mathematically invert the degradation (e.g., invert the blurring filter), and *enhancement* refers to the improvement of the overall image quality without explicit mathematical inversion of the degradation process.

Theory

The problems of image enhancement and restoration are ill posed since they amount to recovering some image information that has been eliminated during the degradation. Solving these problems must therefore rely on some prior knowledge of the image, or in mathematical terms *image models*, to regularize the solution. Mathematically, let f denote an ideal image, U a linear degradation operator, w an additive noise, and

$$y = Uf + w \tag{1}$$

the degraded (observed) image. While this model does not cover all possible degradation scenarios, it is very popular and useful, and serves to illustrate the underlying image enhancement and restoration key concepts. Modern image enhancement and restoration estimates the underlying image f from the degraded observation y by, for example, minimizing a functional of the form

$$\hat{f} = \arg\min_{h} \left(\|y - Uh\|^2 + \varphi(h) \right), \tag{2}$$

where the first term ensures that the restored image $\hat{f}$ and the degraded image y agree with the image degradation (Eq. 1), and the second term $\varphi(h)$ regularizes the solution via a certain image model. The technology of image enhancement and restoration thus has been developed hand in hand with a better understanding of image modeling.

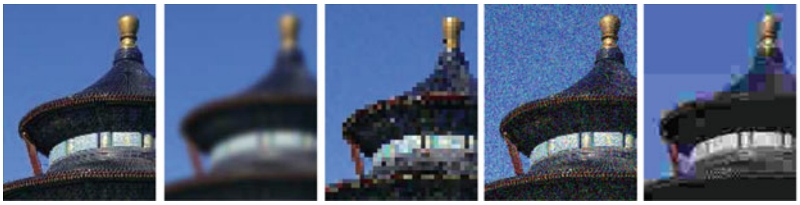


Image Enhancement and Restoration, Fig. 1 From *left* to *right*. Ideal image, image degraded by out-of-focus, resolution downgrade, noise spots, and JPEG block artifact

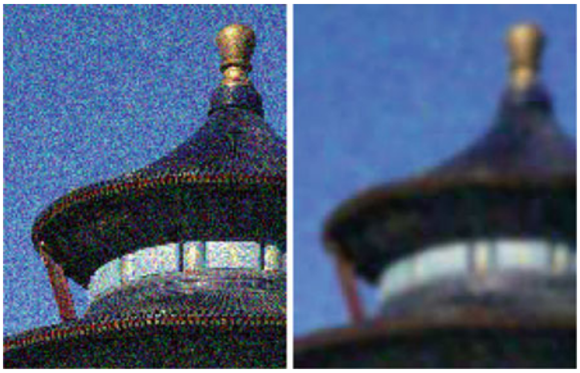


Image Enhancement and Restoration, Fig. 2 *Left*: noisy image. *Right*: image denoised by Gaussian smoothing

The most classic image model that dated from the 1960s assumes that image content is uniformly smooth [10]. This model results in a number of well-known image enhancement and restoration algorithms, including Gaussian smoothing for denoising, bicubic interpolation for zooming, and Wiener filter for deblurring [12]. All these algorithms are implemented with *linear* filtering uniformly applied over the image, typical isotropic local filters smoothing out the image. While the uniformly smooth assumption holds on regular image regions such as sky or a blackboard, that typically dominates a natural image, it is obviously oversimplified on other important types of image transition structures, such as contours, that are smooth along one direction but not the other, and textures that are oscillatory patterns. As shown in Fig. 2, although image noise is attenuated, image contours become blurred at the end of restoration when this simple uniformly smooth model is assumed.

Anisotropic image models attempting to address this problem came into the scene in the early 1990s (with some works dating to the 1960s as well, by Gabor). As opposed to the uniformly smooth assumption, the anisotropic models assume that an image is piecewise smooth, in other words, smooth inside each sub-region, and that at a contour or boundary of the regions where the image intensity sharply changes, the smoothness holds only along the contour direction but not in the perpendicular direction. These models give clearly a better image description and have been elegantly formulated in some partial differential equation frameworks such as anisotropic diffusion [7, 15] and total variation [16]. The resulting algorithms implement *nonlinear* filtering adaptive to the image content, uniformly smoothing inside each image sub-region, and smoothing only along the contour direction on the region boundaries. Therefore, image contours are better preserved.

Since the boom of wavelets in the early 1990s, multi-resolution harmonic analysis has lead to considerable efforts and improvements on image modeling and restoration [4, 12]. Wavelet analysis models an image from multiple resolutions; at each resolution, translating local wavelet atoms oscillating at the corresponding scale are used. The wavelet response is typically high on image transition structures, such as contours and textures, and negligible on regular regions. As a result, it does not only implement *nonlinear* adaptive filtering, but also reveals the important concept of "sparse modeling": wavelet analysis represents an image with only a few large wavelet coefficients that absorb most of the image energy, while the majority of wavelet coefficients quickly decay to zero. The wavelet's sparsity as well as its performance in image enhancement and restoration have been later improved by geometric adaptive harmonic analysis such as curvelets [5] that include local directional atoms to catch the image contours.

In order to further promote the resulting sparsity relative to prefixed harmonic analysis dictionaries (Dictionary here means an ensemble of harmonic analysis atoms), such as wavelets or curvelets, *sparsifying learned dictionaries*, i.e., dictionaries that are learned

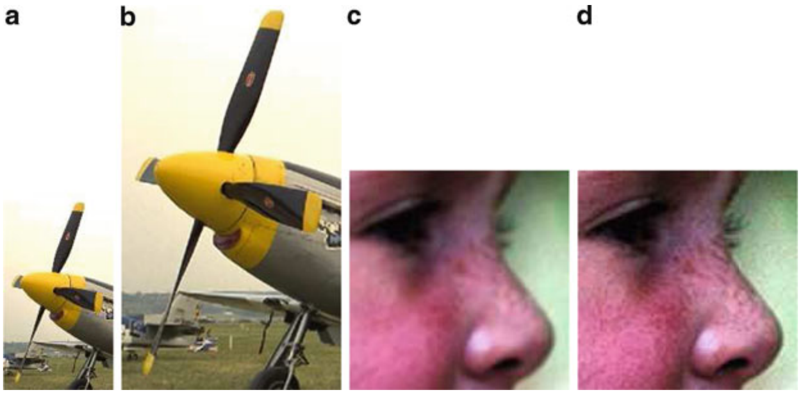


Image Enhancement and Restoration, Fig. 3 Image enhancement and restoration examples. (**a**) and (**b**) Super-resolution zooming: Low-resolution and zoomed images. (**c**) and (**d**) Deblurring: blurred and deblurred images (Figures reproduced from [17])

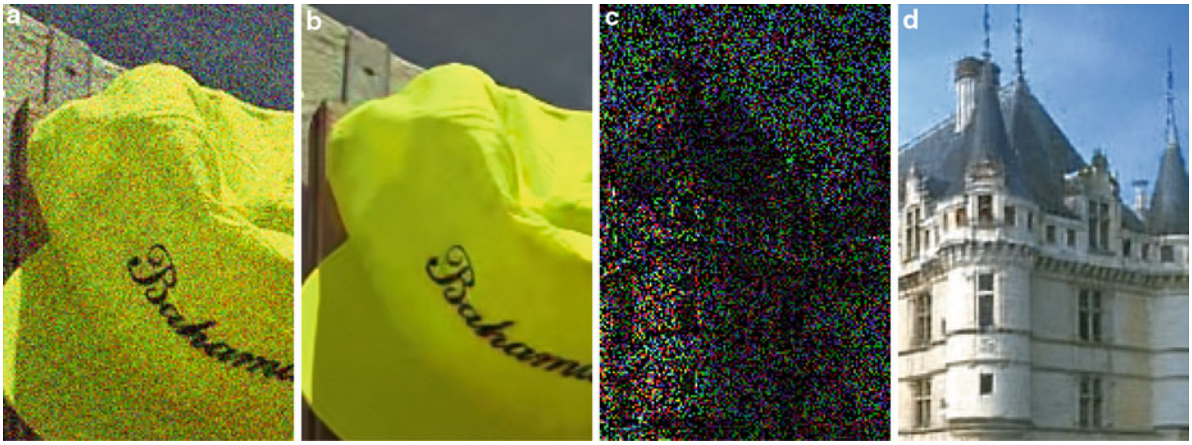


Image Enhancement and Restoration, Fig. 4 Image enhancement and restoration examples. (**a**) and (**b**) Denoising: Noisy and denoised images. (**c**) and (**d**) Inpainting: image with 80 % random missing pixels and restore image. (This problem is related to the task of reconstructing a color image from undersampled color channels, as present in most low/mid, end digital cameras.) The results are obtained following the technique in [17]

from images of interest to yield sparse representations for that class of data, have emerged [1, 13], leading to further improved image enhancement and restoration performance [11].

Non-local image modeling is based on the observation that images typically contain repetitive local patterns (self-similarity). Since the pioneering work of the nonlocal means denoising algorithm [3] in 2005 (see also [2, 14]), non-local modeling has been extensively studied in image enhancement and restoration [8, 9].

Gaussian mixture models, a statistical model widely applied in machine learning, have been shown particularly effective for image enhancement and restoration [17]. The models assume that local image patches follow a mixture of Gaussian distributions. The resulting *piecewise linear* algorithm is not only extremely fast, but also reveals some connections to sparse modeling and non-local modeling.

State-of-the-art image enhancement and restoration results are obtained with algorithms derived from the last three image models, namely, sparse modeling with learned dictionaries, non-local modeling, and Gaussian mixture models. Figure 3 illustrates some examples.

Open Problems

For image enhancement and restoration problems such as removing Gaussian white noise from an image and filling small holes at random positions in an image, it seems that the current performance is already acceptable, as illustrated in Fig. 4, and has arguably reached a quality boundary uneasy to go beyond. For other more difficult problems such as deblurring and zooming, although substantial visual quality improvement has been achieved with respect to classic algorithms such as Wiener filter and bicubic interpolation, objective performance improvement is relatively limited despite considerable efforts that have been devoted. Theoretical performance bounds of image enhancement and restoration remains to be understood. The recent very exciting compressive sensing theory [6] reveals the performance bounds of the sparse modeling approaches given some random degradation operations, but is inapplicable to typical degradations such as blurring and subsampling and to the most successful learned dictionaries. The extension of these results to more realistic image degradation scenarios and image models is among the current challenges of image restoration and enhancement.

References

1. Aharon M, Elad M, Bruckstein A (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311
2. Awate SP, Whitaker RT (2005) Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering. In: Proceedings of conference on computer vision and pattern recognition (CVPR), vol 2, San Diego, pp 44–51
3. Buades A, Coll B, Morel JM (2006) A review of image denoising algorithms, with a new one. Multiscale Modeling Simul 4(2):490–530
4. Burt P, Adelson E (1983) The Laplacian pyramid as a compact image code. IEEE Trans Commun 31(4):532–540
5. Candes EJ, Donoho DL (2004) New tight frames of curvelets and optimal representations of objects with C2 singularities. Commun Pure Appl Math 56:219–266
6. Candès EJ, Tao T (2006) Near-optimal signal recovery from random projections: Universal encoding strategies? IEEE Trans Inf Theory 52(12):5406–5425
7. Catté F, Lions PL, Morel JM, Coll T (1992) Image selective smoothing and edge detection by nonlinear diffusion. SIAM J Numer Anal 29(1):182–193
8. Dabov K, Foi A, Katkovnik V, Egiazarian K (2007) Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans Image Process 16(8):2080–2095
9. Gilboa G, Osher S (2008) Nonlocal operators with applications to image processing. Multiscale Modeling Simul 7(3):1005–1028
10. Lindenbaum M, Fischer M, Bruckstein A (1994) On Gabor's contribution to image enhancement. Pattern Recognit 27(1):1–8
11. Mairal J, Elad M, Sapiro G (2007) Sparse representation for color image restoration. IEEE Trans Image Process 17(1):53–69
12. Mallat S (2009) A wavelet tour of signal processing: the sparse way, 3rd edn. Academic, Burlington
13. Olshausen BA, Field DJ (1996) Natural image statistics and efficient coding*. Netw Comput Neural Syst 7(2):333–339
14. Ordentlich E, Seroussi G, Verdu S, Weinberger M, Weissman T (2003) A discrete universal denoiser and its application to binary images. In: Proceedings of international conference on image processing, vol 1, Barcelona
15. Perona P, Malik J (1990) Scale-space and edge detection using anisotropic diffusion. IEEE Trans Pattern Anal Mach Intell 12(7):629–639
16. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. Phys D 60(1–4): 259–268
17. Yu G, Sapiro G, Mallat S (2010) Solving inverse problems with piecewise linear estimators: from Gaussian mixture models to structured sparsity. Image Processing, IEEE Transactions 2481–2499

Image Inverse Problems

►Image Enhancement and Restoration

Image Mosaicing

►Image Stitching

Image Plane

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

Synonyms

Retina

Related Concepts

►Pinhole Camera Model

Definition

The image plane is the planar surface on which the image is generated in an image formation process or a model thereof.

Background

In most cameras, the photosensitive elements are arranged on a planar support. In image formation models, the image plane is the (mathematical) plane where the image is formed and within which pixels or film are supposed to be located.

There exist cameras where the photosensitive area is not flat. For instance, in most early panoramic image acquisition systems that proceeded by scanning a scene with a rotating slit camera, the film was wrapped onto the inside of a cylindrical surface [1, 2]. In that case, one may still devise an equivalent theoretical image formation model that has a planar image support surface.

References

1. McBride B (2011) A timeline of panoramic cameras. http://www.panoramicphoto.com/timeline.htm Accessed 3 August 2011
2. Benosman R, Kang S (2001) A brief historical perspective on panorama. In: Benosman R, Kang S (eds): Panoramic vision: sensors, theory, and applications. Springer, Verlag, pp 5–20

Image Registration

Daniel C. Alexander
Centre for Medical Image Computing, Department of Computer Science, University College London, London, UK

Synonyms

Image alignment

Definition

Image registration aligns corresponding features of images via spatial transformations.

Background

Computer vision or image processing systems often need to align multiple images of the same or similar scenes. In medical imaging, for example, radiologists routinely compare images of a patient acquired at different times to monitor changes. The intensity difference between two images highlights such changes but only if the corresponding features are in the same location. However, patients' positions in imaging devices vary between visits, so raw images never have perfect alignment. Image registration transforms or warps one image so that the important objects and regions are in the same position as in the other image. The difference image then reveals intrinsic physical changes. Figure 1 illustrates the idea. The problem becomes more challenging when the images come from different devices (inter-modality registration) or from different subjects (intersubject registration).

The same problem arises in nonmedical imaging applications. Surveillance systems, for example, often need to look for differences between images at different times, for example, to subtract the background and highlight activity in a scene viewed by a security camera. Fixed cameras can wobble in the wind and produce misaligned images that require registration before the difference image provides a meaningful result. Stitching images together to create panoramas [1–3] also requires image registration to align the overlapping parts of the images being stitched together; Fig. 2 illustrates this application. Similarly, super-resolution techniques [4] align multiple images of the same scene and infer subpixel detail.

Theory

The process of automatic image registration involves optimizing a cost function, which expresses the similarity of the two images, with respect to the parameters of a transformation of one of the images. Mathematically, the optimization problem is

$$\{T^\star, g^\star\} = \underset{\{T,g\}}{\operatorname{argmin}}(f(I_1, T(g(I_2)))), \qquad (1)$$

where I_1 is the target image, which is fixed; I_2 is the source image, which the transformations T and g act upon; T is a spatial transformation, or warp; and g affects only the image intensity at each pixel position; the optimization seeks the transformations $T^\star$ and $g^\star$ that minimize the cost function f.

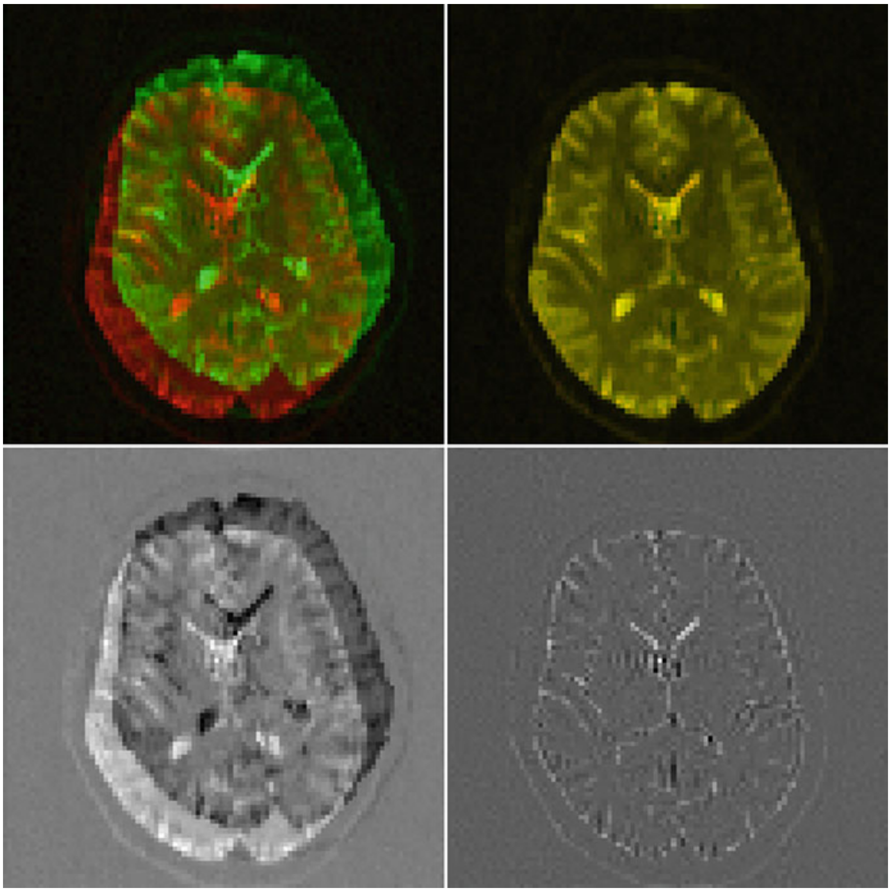


Image Registration, Fig. 1 Intrasubject brain image registration. *Top left*: overlaid images of the same brain from different acquisitions, one in *red* and one in *green*; *bottom left*: difference image of the two unaligned images; *top right*: overlaid images after registration via a rigid transformation; *bottom right*: difference image after alignment

The process decomposes into four key components, which the following subsections discuss one by one. The literature contains many review papers, for example, [5–8], that discuss each component in more detail.

Features

Various image features can drive the registration process. Broadly, the feature set is either sparse or dense.

Sparse feature sets consist of geometric features identified in the image through some preprocessing step. These features might be salient points identified by a user or by an automatic detector; SIFT features [9] or variants thereof are a common choice. Features may also be more complex geometric objects, such as salient lines, curves, surfaces, or regions.

Dense features are typically pixel-by-pixel image intensities. Each feature may be a single scalar intensity or may have multiple components, as in multispectral images. Dense feature sets may not include every pixel in the image and often exclude pixels that lie outside the salient region of the image, such as the brain region in Fig. 1.

Cost Function

The cost function provides a measurement of similarity between two images. The definition of similarity depends on the set of features.

For sparse feature sets, the cost function typically uses a measure of distance between matched features in the two images. For example, if the feature set is a list of salient points in each image, the Euclidean distance between each corresponding pair of points

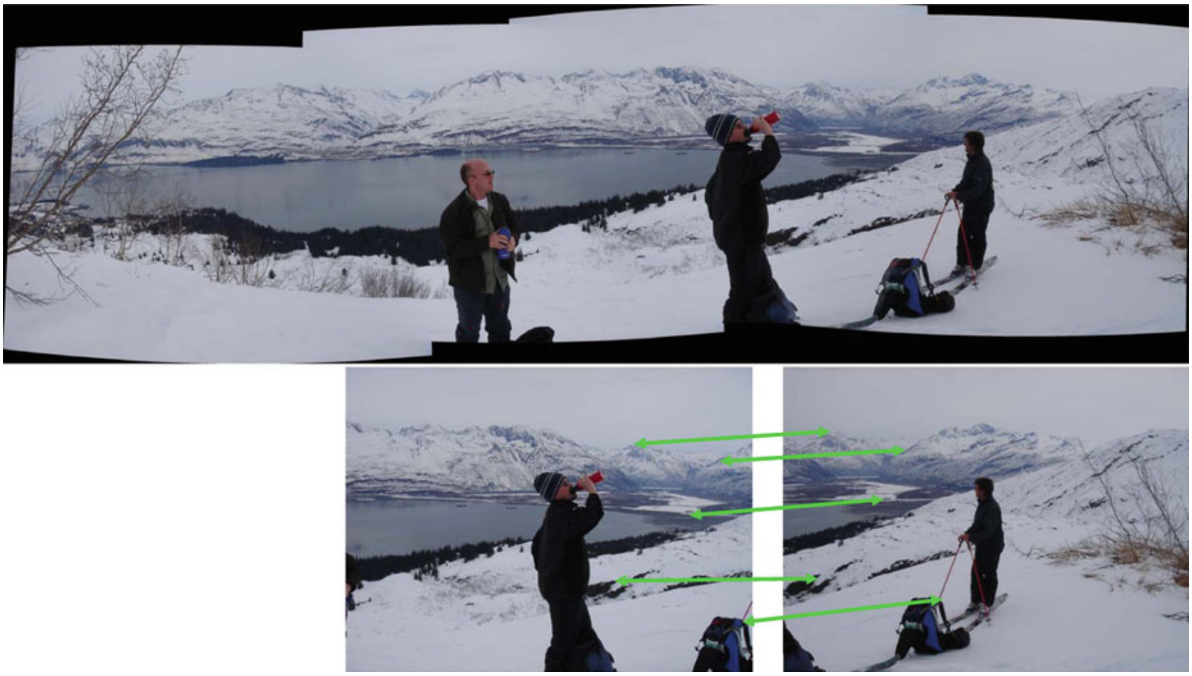


Image Registration, Fig. 2 Image registration for stitching. The panoramic image at the *top* comes from stitching together various images including the two at the *bottom*. Image registration provides the spatial transformation that associates corresponding salient points in the two images, such as those marked by the *green arrows*

provides a measure of similarity. This requires a preceding step to establish correspondence between pairs of points, in a similar way to various other computer vision tasks, such as stereo matching.

Registration based on dense features, that is, pixel intensities, typically uses statistical measures of similarity between pixel intensities in corresponding locations. The most direct measure of similarity uses the average intensity difference

$$f_1(I_1, I_2) = -\sum_{x \in X} |I_1(x) - I_2(x)|, \qquad (2)$$

where X is the salient set of pixels. Equation 2 uses the L_1 norm, but other norms are equally possible.

Direct intensity comparisons, as in f_1, assume that the pixel intensity at corresponding locations is the same subject to some noise perturbation. However, that assumption often does not hold. For example, differences in intensity scale between images arise frequently. Where such intensity differences are likely, similarity measures based on the correlation of pixel intensities between the two images are more appropriate. For example, the normalized cross correlation

$$f_2(I_1, I_2) = \frac{\sum_{x \in X}(I_1(x) - \bar{I}_1)(I_2(x) - \bar{I}_2)}{\sqrt{\sum_{x \in X}(I_1(x) - \bar{I}_1)^2 \sum_{x \in X}(I_2(x) - \bar{I}_2)^2}}, \qquad (3)$$

where $\bar{I}$ is the mean intensity of image I over region X.

The mapping between intensities of corresponding pixels is sometimes more complex than a simple scale change. It may be nonlinear and non-monotonic. For example, in inter-modality medical image registration, two images of the same object may have the same regional structure but different regional contrast: image 1 has higher intensity than image 2 in some regions, vice versa in others, and intensity correlation at alignment remains low. Entropy-based similarity measures [10–13] provide a useful alternative. A common choice is the normalized mutual information

$$f_3(I_1, I_2) = \frac{H(I_1) + H(I_2)}{H(I_1, I_2)}, \qquad (4)$$

where

$$H(I) = -\sum_{x \in X} p(I(x)) \log p(I(x)) \tag{5}$$

is the entropy of image I, with p the distribution of image intensities in I, and

$$H(I_1, I_2) = -\sum_{x \in X} p(I_1(x), I_2(x)) \log p(I_1(x), I_2(x)) \tag{6}$$

is the joint entropy of images I_1 and I_2, with p now the joint intensity distribution. The cost function f_3 is minimum when one image predicts the other most parsimoniously, that is, when the intensity mapping from I_1 to I_2 requires the least information to describe. Mutual-information-based cost functions prove remarkably effective and robust. In practice, they are often preferred to direct comparison or correlation-based cost functions, such as f_1 and f_2, even for intrasubject intra-modality registration.

Transformation Model

A variety of models are available for the spatial transformation, T. Simple transformations, such as rigid, affine, or polynomial transformations, are global in the sense that even well-separated pixels undergo highly correlated displacements. More complex models, such as spline [14], radial-basis-function [15], elastic [16], or fluid [17] transformations, can have more local properties, so that the displacement of one pixel under the transformation correlates only with that of proximal pixels.

In some applications, simple global transformations are sufficient. For example, in brain imaging, rigid transformations are often sufficient to align two images from the same subject. Since a rigid skull encases the brain, it deforms very little between image acquisitions. The registration needs to correct only for the difference in position and orientation of the subject in the imaging device. Higher-order global transformations, such as full affine or polynomial transformations, can improve alignment significantly even when the physical transformation is rigid, because they can capture artifactual distortions introduced by the image device. Image stitching often uses a homography, which is a global transformation that accounts for changes in perspective.

Local transformations can capture more subtle changes between images. They are essential, for example, for detecting and quantifying local atrophy (shrinkage) of brain tissue that occurs over time in various neurological conditions [18]. In general, in medical imaging, local transformations are usually necessary for good alignment in intersubject image registration, where local variations in size and shape of organs and body structure arise.

In practice, the intensity transformation, g in Eq. 1, is often the identity. However, g becomes important in images that contain more complex information at each pixel than single or multiple scalar values. For example, vector or tensor images are common in remote sensing and medical imaging. In such images, each pixel has an associated orientation. Nontrivial g is essential to ensure that local orientations remain consistent with the image structure through the spatial transformation; see, for example, [19, 20].

Optimization

The wide range of optimization algorithms available today, from simple line search or gradient descent to stochastic and genetic optimization procedures, provides many candidates for driving the minimization of the cost function that solves the registration problem. The choice of optimization procedure depends on the feature set. Registration via sparse feature matching often relies on algorithms like RANSAC [21], which are robust to errors in point correspondence, whereas most image registration algorithms with dense features use some form of gradient descent. The cost function is almost always non-convex, and an effective optimization procedure for reliable image registration cannot ignore local minima. Even for simple rigid transformations, local minima often arise and reliable rigid registration with gradient descent requires repeated runs with multiple starting points [22]. The optimization problem tends to become harder the more complex the transformation model. In local registration, the optimization has a much larger number of parameters, so takes longer, and repeated runs can be impractical. Hierarchical approaches, which start with a simple global registration to get a good starting point and gradually add parameters and reoptimize, are common to obtain a good local registration. Multi-resolution strategies, which start with low-resolution images and gradually increase resolution, also help.

Application

The medical imaging community is a large consumer and developer of image registration techniques. Intrasubject image registration enables fusion of information in images from different devices. Intrasubject registration also enables tracking of changes over time during development or disease. In drug trials, for example, imaging offers the potential to observe the effects of a prospective treatment and establish its efficacy noninvasively; image registration is essential for monitoring such effects.

Another major application is spatial normalization for group studies, which study the variation in the size, shape, and internal organization of a particular organ or object. A common application is human brain mapping where morphological variability is well studied in a range of conditions. Intersubject image registration ensures that a collection of similar images are in the same spatial frame of reference so that studies of variation are meaningful. This spatial normalization allows, for example, medical imaging researchers to characterize differences in organ size, shape, and structure between different populations, such as normal healthy adults and patients with a certain condition.

Image registration for image stitching enables day-to-day image processing for digital camera users, as standard packages like Photoshop include such operations. Google Maps is a large-scale application of the same technology.

Many implementations of image registration are freely available. Tried and tested global registration software includes the FLIRT package [22] and Nifty Reg [23]. The popular b-spline registration algorithm [14] has implementations with some variations in FNIRT [24] and Nifty Reg [25], which also offers a GPU implementation. The DARTEL package [26] is designed specifically for spatial normalization of large brain image ensembles. The recent ANTS package [27] combines several state-of-the-art ideas and performs well in a head-to-head evaluation with other standard packages [28].

Open Problems

Consistency remains an open problem in image registration. Basic algorithms do not ensure that the transformation from registering images A to B is the perfect inverse of that from registering images B to A. Symmetry constraints on the cost function to ensure binary consistency are straightforward to enforce. However, the problem becomes more complex as the number of images to align increases: ensuring consistency of A to B to C with C to A is more challenging. Groupwise registration, as in [26], goes some way towards ameliorating this problem.

Topological differences or changes present a further open challenge. Most transformation models do not accommodate differences in topology naturally between images. In fact, significant effort has gone into developing diffeomorphic transformation models that cannot fold or tear. However, topological differences arise frequently. In intersubject medical image registration, for example, it is not uncommon for an anatomical feature in one person to be entirely missing in another. The same problem can arise even intrasubject, say, before and after surgery to remove a tumor.

References

1. Szeliski R, Shum HY (1997) Creating full view panoramic image mosaics and environment maps. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques. SIGGRAPH '97, New York, NY, USA. ACM/Addison-Wesley, pp 251–258
2. Szeliski R (2006) Image alignment and stitching: a tutorial. Found Trends Comput Graph Vis 2:1–104
3. Brown M, Lowe D (2007) Automatic panoramic image stitching using invariant features. Int J Comput Vis 74: 59–73. doi:10.1007/s11263-006-0002-3
4. Irani M, Peleg S (1991) Improving resolution by image registration. CVGIP 53(3):231–239
5. Maintz J, Viergever MA (1998) A survey of medical image registration. Med Image Anal 2(1):1–36
6. Lester H, Arridge SR (1999) A survey of hierarchical non-linear medical image registration. Pattern Recognit 32(1):129–149
7. Hajnal JV, Hill DLG, Hawkes DJ (2001) Medical image registration. CRC Press, Boca Raton. ISBN 0-8493-0064-9
8. Zitov B, Flusser J (2003) Image registration methods: a survey. Image Vis Comput 21(11):977–1000
9. Lowe DG (2004) Distinctive Image Features from Scale-Invariant Keypoints. Int J Comput Vision 60(2): 91–110
10. Studholme C, Hill DLG, Hawkes DJ (1995) Multiresolution voxel similarity measures for mr-pet registration. In: Proceedings of the information processing in medical imaging, Kluwer, pp 287–298
11. Collignon A, Maes F, Delaere D, Vandermeulen D, Suetens P, Marchal G (1995) Automated multi-modality image registration based on information theory. In: Proceedings of the information processing in medical imaging, Kluwer, pp 263–274

12. Viola P, Wells WM (1995) Alignment by maximization of mutual information. In: Proceedings of the international conference on computer vision, IEEE, pp 16–23
13. Pluim JPW, Maintz JBA, Viergever MA (2003) Mutual information based registration of medical images: a survey. IEEE Trans Med Imaging 22:986–1004
14. Rueckert D, Sonoda LI, Hayes C, Hill DLG, Leach MO, Hawkes DJ (1999) Non-rigid registration using free-form deformations: application to breast mr images. IEEE Trans Med Imaging 18:712–721
15. Fornefett M, Rohr K, Stiehl HS (2001) Radial basis functions with compact support for elastic registration of medical images. Image Vis Comput 19:87–96
16. Bajcsy R, Kovacic S (1989) Multiresolution elastic matching. Comput Vis Graph Image Process 46(1):1–21
17. Crum WR, Scahill RI, Fox NC (2001) Automated hippocampal segmentation by regional fluid registration of serial MRI: validation and application in alzheimer's disease. NeuroImage 13(5):847–855
18. Scahill RI, Schott JM, Stevens JM, Rossor MN, Fox NC (2002) Mapping the evolution of regional atrophy in alzheimer's disease: unbiased analysis of fluid-registered serial mri. Proc Natl Acad Sci 99:4703–4707
19. Alexander DC, Pierpaoli C, Basser PJ, Gee JC (2001) Spatial transformations of diffusion tensor magnetic resonance images. IEEE Trans Med Imaging 20:1131–1139
20. Zhang H, Yushkevich PA, Alexander DC, Gee JC (2006) Deformable registration of diffusion tensor mr images with explicit orientation optimization. Med Image Anal 10: 764–785
21. Fischler MA, Bolles RC (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24:381–395
22. Jenkinson M, Smith S (2001) A global optimisation method for robust affine registration of brain images. Med Image Anal 5(2):143–156
23. Ourselin S, Roche A, Subsol G, Pennec X, Ayache N (2001) Reconstructing a 3d structure from serial histological sections. Image Vis Comput 19:25–31
24. Andersson J, Smith S, Jenkinson M (2008) FNIRT: FMRIB's non-linear image registration tool. In: 14th annual meeting of the organization for human brain mapping, OHBM
25. Modat M, Ridgway GR, Taylor ZA, Lehmann M, Barnes J, Fox NC, Hawkes DJ, Ourselin S (2009) Fast free-form deformation using graphics processing units. Comput Methods Programs Biomed 98(3):278–284
26. Ashburner J (2007) A fast diffeomorphic image registration algorithm. NeuroImage 38(1):95–113
27. Avants B, Epstein C, Grossman M, Gee J (2008) Symmetric diffeomorphic image registration with cross-correlation: evaluating automated labeling of elderly and neurodegenerative brain. Med Image Anal 12(1):26–41. Special issue on the third international workshop on biomedical image registration – WBIR 2006
28. Klein A, Andersson J, Ardekani BA, Ashburner J, Avants B, Chiang MC, Christensen GE, Collins DL, Gee J, Hellier P, Song JH, Jenkinson M, Lepage C, Rueckert D, Thompson P, Vercauteren T, Woods RP, Mann JJ, Parsey RV (2009) Evaluation of 14 nonlinear deformation algorithms applied to human brain mri registration. NeuroImage 46(3):786–802

Image Stitching

Matthew Brown
Dept of Computer Science, University of Bath,
Bath, UK

Synonyms

Image Mosaicing; Panoramic stitching

Related Concepts

▶Environment Mapping

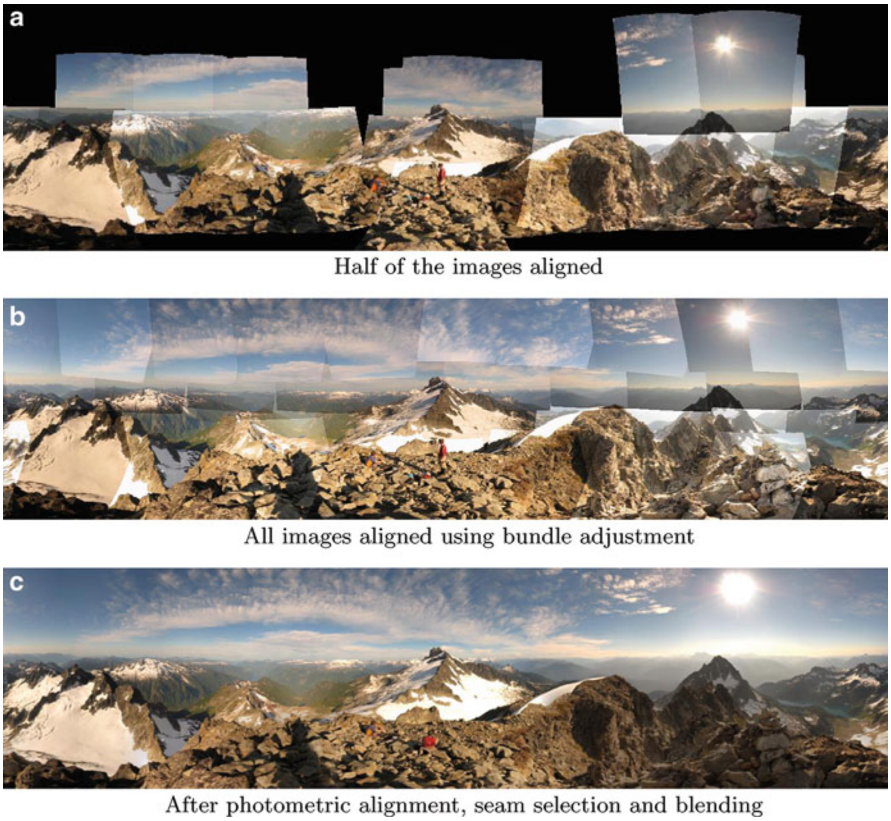
Definition

Image stitching is the process of combining multiple overlapping images to generate a new image with a larger field of view than the originals.

Background

Image stitching can enhance the capabilities of an ordinary camera, enabling the capture of larger field-of-view, higher-resolution images. A popular example is the construction of panorama images by seamlessly combining several images of a scene taken from the same point (This process is known as panoramic stitching, which refers to the special case of image stitching for rotational motion). By capturing images with variable exposure settings, it can also be used to generate images with a higher dynamic range than the originals.

Stitching techniques were originally used in photogrammetry to produce maps from aerial and satellite images. Early techniques involved manual specification of matching images and control points (correspondences) between them [1]. Later methods used automated image alignment [2, 3] and interactive viewers to visualize the results [4]. Modern stitching pipelines offer fully automated operation [3], seam selection [5], and photometric, as well as geometric alignment [6].



Half of the images aligned

All images aligned using bundle adjustment

After photometric alignment, seam selection and blending

Image Stitching, Fig. 1 Panoramic stitching. Images are first geometrically aligned (using a rotational motion model in this case). Photometric alignment is used to compensate for brightness variations between the images, and the final panorama is rendered using seam selection and pyramid blending. (**a**) Half of the images aligned. (**b**) All images aligned using bundle adjustment. (**c**) After photometric alignment, seam selection and blending

A typical pipeline for image stitching consists of the following stages (see Fig. 1):

1. Estimating two-frame motion and discovering overlaps between the images
2. Global alignment (e.g., using bundle adjustment [7])
3. Photometric alignment and seam selection/deghosting
4. Rendering the final panorama with blending and/or tone mapping

Theory and Applications

Image stitching is possible when a one-to-one mapping exists between the source image coordinates. Two commonly occurring examples are: (1) a camera rotating about it's optical center and (2) cameras viewing a planar scene. If the cameras are assumed to be rectilinear, the image coordinates are related by a homography

$$\tilde{\mathbf{u}}_2 = \mathbf{H}_{12}\tilde{\mathbf{u}}_1 \ , \qquad (1)$$

where $\tilde{\mathbf{u}}_1, \tilde{\mathbf{u}}_2$ are the homogeneous coordinates in image 1 and 2 and $\mathbf{H}_{12}$ is a 3×3 matrix that encodes the relative camera positions. For example, in the rotational case, $\mathbf{H}_{12}$ is given by

$$\mathbf{H}_{12} = \mathbf{K}_2\mathbf{R}_2\mathbf{R}_1^T\mathbf{K}_1^{-1} \ , \qquad (2)$$

where $\mathbf{R}_1, \mathbf{R}_2$ are the rotation matrices of cameras 1 and 2 and $\mathbf{K}_1, \mathbf{K}_2$ contain the intrinsic parameters.

A typical image stitching approach begins by robustly estimating $\mathbf{H}_{12}$ from correspondences of local image features [8]. A standard method is to use the RANSAC algorithm [9] to sample the space of transformation hypotheses, for all images with a sufficiently large number of feature matches. One can then reason about the adjacency relationships and recognize panoramas by making a match/no-match decision for each pair and finding connected components in the resulting graph of image matches [3].

After pairwise alignment, gaps and inconsistencies can still exist. Bundle adjustment [7] can be used to minimize projection errors between feature matches in all images and generate globally consistent results. Best results are achieved by parameterizing in terms of the intrinsic and extrinsic parameters of the cameras (e.g., rotation, focal length, radial distortion) [2]. Direct methods [10] (using all of the pixel data instead of only feature points) may optionally be used for accurate final registration.

Once the images are geometrically aligned, the remaining task is to render a seamless output view. The appropriate render surface may depend on the images being aligned: rectilinear renderings (preserving straight lines) might be best for stitching planar surfaces such as whiteboards, spherical or cylindrical render surfaces are popular for wide-angle panoramas. Multiperspective renderings can be used to preserve important geometric properties in the output [11].

Ideally, one can capture or estimate irradiance values per pixel, and given perfect alignment, these would be equal in all images overlapping a given ray. In practice, however, several sources of error contribute toward differences in the recorded radiances. Some common examples are parallax due to motion of the camera center, errors or unmodeled parameters in the camera pose estimate, and moving objects in the scene. Several algorithms have been developed to eliminate the visual seams that result. The best approaches find seam lines which minimize differences between image intensities or radiances [12], and smoothly interpolate between images using pyramid blending [13] or gradient domain fusion [5]. The final results can be tone mapped for display.

An example of an automated capture system capable of stitching gigapixel panoramas with feature-based alignment, seam selection, and dynamic tone mapping is given in [14].

References

1. Slama CC (ed) (1980) Manual of photogrammetry, 4th edn. American Society of Photogrammetry, Falls Church, Virginia
2. Szeliski R, Shum H (1997) Creating full view panoramic image mosaics and environment maps. Comput Graph (SIGGRAPH'97) 31(Annual Conference Series):251–258
3. Brown M, Lowe D (2007) Automatic panoramic image stitching using invariant features. Int J Comput Vis 74(1):59–73
4. Chen S (1995) QuickTime VR – an image-based approach to virtual environment navigation. In: ACM transactions on graphics (SIGGRAPH'95), vol 29, pp 29–38
5. Agarwala A, Dontcheva M, Agarwala M, Drucker S, Colburn A, Curless B, Salesin D, Cohen M (2004) Interactive digital photomontage. In: ACM transactions on graphics (SIGGRAPH'04)
6. Eden A, Uyttendaele M, Szeliski R (2006) Seamless image stitching of scenes with large motions and exposure differences. In: IEEE computer society conference on computer vision and pattern recognition (CVPR'06)
7. Triggs W, McLauchlan P, Hartley R, Fitzgibbon A (1999) Bundle adjustment: a modern synthesis. In: Vision algorithms: theory and practice, number 1883 in LNCS. Springer, Corfu, September 1999, pp 298–373
8. Szeliski R (2010) Computer vision: algorithms and applications, Chapter 9. Springer, http://www.springer.com/computer/image+processing/book/978-1-84882-934-3
9. Fischler M, Bolles R (1981) Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. Commun ACM 24: 381–395
10. Irani M, Anandan P (1999) About direct methods. In: Triggs B, Zisserman A, Szeliski R (eds) Vision algorithms: theory and practice, number 1883 in LNCS. Springer, Corfu, September 1999, pp 267–277
11. Zelnik-Manor L, Peters G, Perona P (2005) Squaring the circle in panoramas. In: Tenth IEEE international conference on computer vision (ICCV'05), Beijing, pp 1292–1299
12. Davis J (1998) Mosaics of scenes with moving objects. In: IEEE computer society conference on computer vision and pattern recognition (CVPR'98), pp 354–360
13. Burt P, Adelson E (1983) A multiresolution spline with application to image mosaics. ACM Trans Graph 2(4): 217–236
14. Kopf J, Uyttendaele M, Deussen O, Cohen M (2007) Capturing and viewing gigapixel images. In: ACM transactions on graphics (SIGGRAPH'07), vol 26

Image-Based Lighting

Tien-Tsin Wong
Department of Computer Science and Engineering,
The Chinese University of Hong Kong, Hong Kong
SAR, China

Synonyms

Environment mapping; Reflection mapping

Related Concepts

▸Plenoptic Function

Definition

Image-based lighting [1, 2] is a rendering technique to compute the reflection from a 3D object lit in a distant environment, represented as an image, typically in the form of a cubemap (Fig. 1).

Background

Due to the computational expense of global illumination (e.g., radiosity and Monte Carlo ray tracing), most real-time graphics systems are depth-buffering based and only support local illumination. This hurts the realism of the rendered images. Environment mapping [1, 3] is proposed to simulate the reflection of the surrounding environment on an object surface (Fig. 1). The enclosing environment is assumed to be infinitely far away (distant environment) because all surface points on the object are assumed to be lit by the same environment. Due to the high computational expense, early implementations of environment mapping account only for the light contribution along the mirror reflection direction. Hence, most surfaces rendered by the environment mapping are over-shiny.

Image-based lighting can be regarded as a more comprehensive realization of the environment mapping, by accounting not only the light contribution along the mirror reflection but also the whole enclosing sphere. The surface reflectance property (bidirectional reflectance distribution function, BRDF) is also considered so as to render not only shiny or glossy surfaces but also most kinds of surfaces. Moreover, the environment maps are usually captured as high dynamic range (HDR) images to further increase the photorealism. Note that the image-based lighting remains work even low-dynamic range (LDR) environment maps are used instead.

Theory

Extending from accounting only the mirror reflection direction to the whole enclosing sphere drastically increases the computational expense. Hence, the challenge is how to efficiently compute the following integration for each surface point:

$$I(x, s) = \int_{\Omega} L_{\text{in}}(\omega)\rho(x, \omega, u)v(x, \omega)(\omega \cdot n)d\omega \quad (1)$$

where x is the current surface point of interest; Ω is the surrounding environment (the distant environment map); ω is the incoming light direction; u is the viewing direction from x towards the eye; I is the reflected light; $L_{\text{in}}(\omega)$ is the incoming light contribution along direction ω, in other words, a point in the environment map; v is the visibility function; ρ is the BRDF; and n is the surface normal at x. Note that L_{in}, ρ, and v are spherical functions.

One way to evaluate the above integration is to approximate the environment map by a much smaller number (say m) of point light sources. The position and color of the point lights are obtained by importance sampling of the environment map [4]. In other words, the above integration is approximated by a summation of light contribution of m point lights. The rendered image can simply be generated by adding m images, each rendered by illuminating the scene with a point light source.

By adopting the image-based relighting techniques [5], the above integration can be evaluated more efficiently. The idea is to first encode the spherical function with basis functions. This effectively converts a huge spherical function (table) into a coefficient vector s_i as follows. Since the basis functions B_i are known, they need not be stored:

$$S(\omega) \approx \sum_{i}^{k} s_i B_i(\omega) \quad (2)$$

where S is a spherical function and k is the total number of basis functions, which is much smaller than the number of entries in the original spherical table.

By embedding $\rho(x, \omega, u)v(x, \omega)(\omega \cdot n)$ into a spherical function C, both C and L_{in} can be encoded with the same basis and stored as two coefficient vectors, c_i and l_i, respectively. If the selected basis functions are orthonormal, the above integration (Eq. 1) can be simply evaluated as a dot product between two coefficient vectors [1],

$$I \approx \sum_{i}^{k} c_i l_i \quad (3)$$

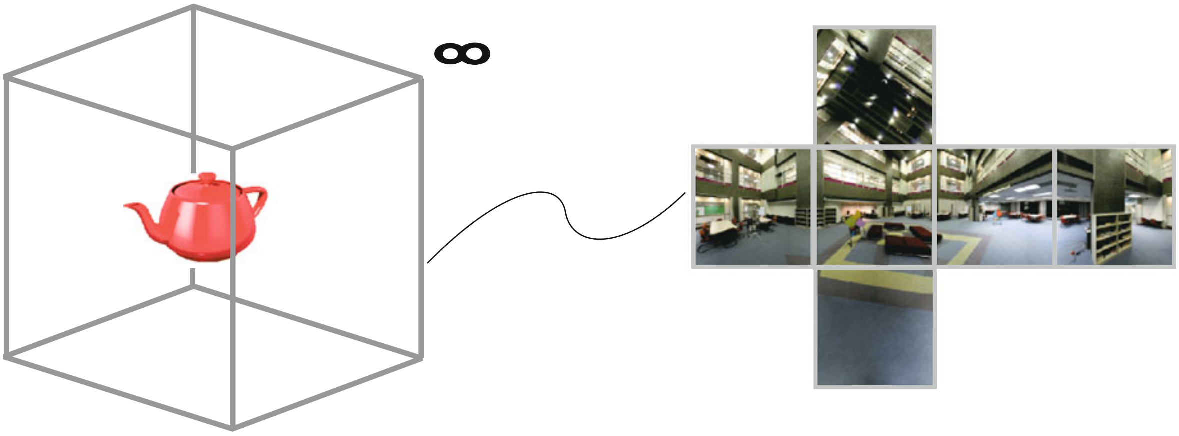


Image-Based Lighting, Fig. 1 Image-based lighting assumes the object being rendered is enclosed by an image-based environment positioned infinitely far away, or equivalently, the object is infinitesimally small. The environment map is typically represented as a cubemap

Basis Functions

The key to efficient image-based lighting is to select an appropriate basis for representing the spherical functions. Various bases have been proposed for image-based relighting [5]. They are directly applicable to image-based lighting.

A pioneer work is proposed by Nimeroff et al. [6]. They efficiently relit the scene under various natural illumination (overcast or clear skylight). The illumination function is decomposed into a linear combination of *steerable functions*.

Principal component analysis is naturally a potential choice for basis function [7]. Singular value decomposition can be used to extract a set of eigen-images from the input reference images. The desired image can then be synthesized by a linear combination of these basis images given a set of coefficients [8, 9] if all surfaces are Lambertian.

Earlier works do not consider the spherical nature of the illumination computation. Wong et al. [5, 10] chose the *spherical harmonic basis*, which is commonly used for compressing BRDF. Pleasant rendering results are obtained with 16–25 basis functions. However, spherical harmonic is also well known in over-smoothing the high-frequency signal (e.g., shadow) in the original spherical function, leading to low-frequency results.

To achieve all-frequency rendering, Haar *wavelet basis* is proposed [11]. It may introduce visual artifact when the distant environment contains a dominant but small-size spot. The cause of such artifact is due to the digitization of the spherical function and the limited reconstruction involving only important wavelet coefficients.

Spherical radial basis function (SRBF) [12–14] is another approach to capture all-frequency signal. The local support nature of SRBF allows its implementation to be very efficient and simple. The multiscale spherical radial basis function [15] avoids the visual artifact of the Haar wavelet basis while remains able to achieve all-frequency rendering.

Application

Image-based lighting can be applied to produce realistic rendering for both off-line movie production or real-time computer games. The parallel nature of image-based lighting (all surface points have to evaluate Eq. 1 independently) facilitates its real-time realization on modern graphics processing unit (GPU).

References

1. Cabral B, Max N, Springmeyer R (1987) Bidirectional reflection functions from surface bump maps. In: Proceedings of the 14th annual conference on computer graphics and interactive techniques. ACM, New York, pp 273–281

2. Debevec P (2002) Image-based lighting. IEEE Comput Graphics Appl 22(2):26–34
3. Blinn JF, Newell ME (1976) Texture and reflection in computer generated images. Commun ACM 19(10):542–547
4. Agarwal S, Ramamoorthi R, Belongie S, Jensen HW (2003) Structured importance sampling of environment maps. ACM Trans Graphics 22(3):605–612
5. Wong TT, Heng PA, Or SH, Ng WY (1997) Image-based rendering with controllable illumination. In: Proceedings of the 8th Eurographics workshop on rendering (Rendering techniques'97), St. Etienne, pp 13–22
6. Nimeroff JS, Simoncelli E, Dorsey J (1994) Efficient re-rendering of naturally illuminated environments. In: Fifth Eurographics workshop on rendering, Darmstadt, Germany, pp 359–373
7. Ho PM, Wong TT, Choy KH, Leung CS (2003) PCA-based compression for image-based relighting. In: Proceedings of IEEE international conference on multimedia and expo 2003 (ICME 2003), vol I, Baltimore, Maryland, USA,pp 473–476
8. Belhumeur PN, Kriegman DJ (1996) What is the set of images of an object under all possible lighting conditions. In: Proceedings of IEEE conference on computer vision and pattern recognition (CVPR), San Francisco
9. Zhang Z (1998) Modeling geometric structure and illumination variation of a scene from real images. In: Proceedings of the international conference on computer vision (ICCV'98), Bombay, India
10. Wong TT, Fu CW, Heng PA, Leung CS (2002) The plenoptic illumination function. IEEE Trans Multimedia 4(3): 361–371
11. Ng R, Ramamoorthi R, Hanrahan P (2003) All-frequency shadows using non-linear wavelet lighting approximation. ACM Trans Graphics 22(3):376–381
12. Wong TT, Leung CS, Choy KH (2005) Lighting precomputation using the relighting map. In: Engel W (ed) ShaderX^3: advanced rendering with DirectX and OpenGL. Charles Rivers Media, Hingham, pp 379–392
13. Leung CS, Wong TT, Lam PM, Choy KH (2006) An RBF-based image compression method for image-based rendering. IEEE Trans Image Process 15(4):1031–1041
14. Tsai YT, Shih ZC (2006) All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. In: Proceedings of ACM SIGGRAPH 2006, Boston, pp 967–976
15. Lam PM, Ho TY, Leung CS, Wong TT (2010) All-frequency lighting with multiscale spherical radial basis functions. IEEE Trans Visual Comput Graphics 16(1):43–56

Image-Based Modeling

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

Synonyms

Three dimensional Modeling from Images

Definition

Image-based modeling refers to the process of using two-dimensional images to create three-dimensional models. These models often consist of a geometric shape and a texture map defined over this shape.

Background

Three dimensional models are mathematical representations of three dimensional objects or scenes. These models are useful for various applications such as simulation, robotics, virtual reality, and digital entertainment. Automatically creating these models has been an important research topic since the early days of computer vision.

Broadly speaking, there are two schools of image-based modeling methods. One group employs range sensor for shape modeling, while the other group uses pure images with binocular or multi-view stereo. One of the origins of the first group is [1], which developed a technique to automatically generate a virtual three dimensional model by observing actual objects along the line of physics-based paradigm. Recently, this direction of research has been accelerated by the development of handy range sensor, such as the Microsoft Kinect and other consumer depth cameras. A representative work from the other group of pure image-based method is [2], which introduced an interactive method to model architectural scenes by fitting geometric primitives to the input images. This direction of research is later generalized to model more general objects. Typically, a cloud of 3D points is first recovered from the input images. A detailed and precise shape representation is then derived from these points, and a texture map is created to represent the color of each point on the shape. This process is illustrated in Figure 1.

The process of obtaining 3D points from input images is known as 3D reconstruction. It is a well-studied problem, and most of the relevant theoretic results are summarized in this handbook [4]. There are also a number of well-established 3D reconstruction software systems such as [5–7]. Though the reconstructed 3D points can be used directly for certain measurements, most of applications require polygonal meshes, NURBS surfaces, or solid shape models. 3D modeling is the process of creating these shape models

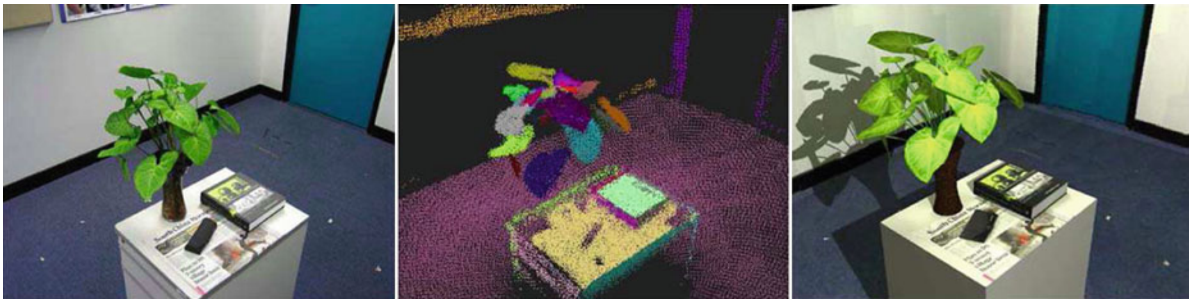


Image-Based Modeling, Fig. 1 A typical pipeline of image-based modeling. In the *left* is one of the input image. In the *middle* is the set of reconstructed 3D points. In the *right* is a rendering of the recovered 3D model of the scene (These pictures are from [3])

from 3D points and 2D images. This modeling process can be automatic or interactive. The prior knowledge of the shape to be modeled often plays an important role in this process. A comprehensive review of recent modeling techniques can be found in [8].

In image-based modeling, the appearance of a shape is often modeled by specifying a color for each point on the shape. These colors can be stored in a so-called texture map. Essentially, this representation assumes the surface is Lambertian and its Bidirectional Reflectance Distribution Function (BRDF) is a constant, that is, the color. In comparison, appearance modeling dedicates to model both the shape and precise surface reflectance properties. More details of appearance modeling can be found in [9].

Theory

In this short chapter, we only focus on the process of creating shape models from 3D points and 2D images. One way to generate these shape models is to obtain a minimal surface automatically from the input data. The minimal surface is a surface that minimizes a functional of the following form

$$\int\int w ds.$$

Here, ds is the infinitesimal surface element and w is the consistency of the surface according to the input 3D points and 2D images. This consistency can be simply the Euclidean distance between the surface and the set of 3D points. Given a set of points $\mathcal{P}$, Zhao et al. [10] defined w as $d(\mathbf{x}, \mathcal{P})$ which is the smallest distance between a surface point $\mathbf{x}$ and other points in the set $\mathcal{P}$. This functional is then minimized by the level set method [11]. Faugeras and Keriven [12] defined w as a function of both the surface position $\mathbf{x}$ and its normal direction $\mathbf{n}$ to facilitate the surface modeling. Lhuillier and Quan [13] further incorporated image reprojection errors and silhouettes in this function w.

The minimal surface-based approach works well when the points are dense and the surface is smooth. However, it has difficulties to model discontinuous surfaces such as hair fibers (linear structure), clothes (open surface patches), tree branches (tree and fractal structure), or buildings (regular axis aligned boxes). There are many existing methods which are designed to exploit the prior knowledge of a shape to facilitate modeling. Wei et al. [14] modeled hair by "growing" 3D smooth curves guided by 3D points and images. Bhat et al. [15] used videos to obtain the parameters of a cloth simulation system. Tan et al. [16] recovered some basic branch elements from the 3D points and used them to generate a fractal branch structure. Xiao et al. [17, 18] identified building facades and repetitive structures on these facades to model streets. Furukawa et al. [19, 20] assumed the scene consists of mutual orthogonal planes at different depth to model buildings. Though generating good results, these methods are limited to model the type of surface that matches their underlying prior shape assumption. A general modeling method is still missing to handle all these different data in a unified framework.

Application

Image-based modeling can be applied in autonomous robotics to generate a three-dimensional map of their environment for path/action planning. It can also be used in industry vision for product quality inspection. The three-dimensional models can also be applied in digital entertainment such as games and movies. Recently, Google Earth and Microsoft Virtual Earth start to provide 3D map services, which can be a very good test bed of large-scale image-based modeling techniques.

References

1. Sato Y, Wheeler MD, Ikeuchi K (1997) Object shape and reflectance modeling from observation. Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '97, New York, NY, USA, ACM Press/Addison-Wesley Publishing Co, pp 379–387
2. Debevec PE, Taylor CJ, Malik J (1996) Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. SIGGRAPH '96, New York, NY, USA, ACM, pp 11–20
3. Quan L, Wang J, Tan P, Yuan L (2007) Image-based modeling by joint segmentation. Int J Comput Vis 75:135–150
4. Hartley R, Zisserman A (2003) Multiple view geometry in computer vision, 2 edn. Cambridge University Press, New York
5. Lhuillier M, Quan L (2005) A quasi-dense approach to surface reconstruction from uncalibrated images. IEEE Trans Pattern Anal Mach Intell 27:418–433
6. Furukawa Y, Ponce J (2010) Accurate, dense, and robust multiview stereopsis. IEEE Trans Pattern Anal Mach Intell 32:1362–1376
7. Snavely N, Seitz SM, Szeliski R (2006) Photo tourism: exploring photo collections in 3d. ACM Trans Graph 25: 835–846
8. Quan L (2010) Image-based modeling, 1 edn. Springer, New York
9. Weyrich T, Lawrence J, Lensch HPA, Rusinkiewicz S, Zickler T (2009) Principles of appearance acquisition and representation. Found Trends Comput Graph Vis 4:75–191
10. Zhao HK, Osher S, Fedkiw R (2001) Fast surface reconstruction using the level set method. In: Proceedings of the IEEE workshop on variational and level set methods (VLSM'01). VLSM '01, Washington, DC, USA. IEEE Computer Society, p 194
11. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. J Comput Phys 79:12–49
12. Faugeras OD, Keriven R (1998) Complete dense stereovision using level set methods. In: Proceedings of the 5th European conference on computer vision-volume i – volume i (ECCV '98). London, UK. Springer, pp 379–393
13. Lhuillier M, Quan L (2003) Surface reconstruction by integrating 3d and 2d data of multiple views. In: Proceedings of the 9th IEEE international conference on computer vision – volume 2. ICCV '03, Washington, DC, USA. IEEE Computer Society, p 1313
14. Wei Y, Ofek E, Quan L, Shum HY (2005) Modeling hair from multiple views. ACM Trans Graph 24:816–820
15. Bhat KS, Twigg CD, Hodgins JK, Khosla PK, Popović Z, Seitz SM (2003) Estimating cloth simulation parameters from video. In: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on computer animation. SCA '03, Aire-la-Ville, Switzerland. Eurographics Association, pp 37–51
16. Tan P, Zeng G, Wang J, Kang SB, Quan L (2007) Image-based tree modeling. ACM Trans Graph 26
17. Xiao J, Fang T, Tan P, Zhao P, Ofek E, Quan L (2008) Image-based facade modeling. ACM Trans Graph 27: 161:1–161:10
18. Xiao J, Fang T, Zhao P, Lhuillier M, Quan L (2009) Image-based street-side city modeling. ACM Trans Graph 28:114: 1–114:12
19. Furukawa Y, Curless B, Seitz S, Szeliski R (2009) Manhattan-world stereo. In: Proceedings of the IEEE international conference on computer vision and pattern recognition (CVPR '09). IEEE Computer Society
20. Furukawa Y, Curless B, Seitz S, Szeliski R (2009) Reconstructing building interiors from images. In: Proceedings of the IEEE international conference on computer vision. ICCV '09. IEEE Computer Society

Image-Based Rendering

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

Synonyms

Image-based rendering (IBR)

Related Concepts

►Light Field; ►Lumigraph; ►Plenoptic Function

Definition

Image-based rendering (IBR) refers to a collection of techniques and representations that allows 3D scenes and objects to be visualized and manipulated in a realistic way without full 3D model reconstruction.

Background

One of the primary goals in computer graphics is photorealistic rendering. Motivated by the difficulties in achieving full photorealism with conventional 3D and model-based graphics, image-based rendering which works directly with real images has proposed as an alternative approach to reduce the rendering and capturing complexity. Depending on how the images are being taken and the auxiliary information, such as depths, etc., required, a number of image-based representations supporting different viewing freedom and functionalities are available. These range from the familiar two-dimension (2D) panoramas to more sophisticated representations such as the four-dimension (4D) light fields [9], lumigraphs [8], and variants, which are special cases of the radiance received at every viewing position, visual angle, wavelength, and time, called the plenoptic function.

The rendering of novel views can therefore be viewed as the reconstruction of the plenoptic function from its samples. Image-based representations are usually densely sampled high-dimensional data with large data sizes, but their samples are highly correlated. Because of the multidimensional nature of image-based representations and scene geometry, much research has been devoted to the efficient capturing, sampling, rendering, and compression of IBR.

Theory

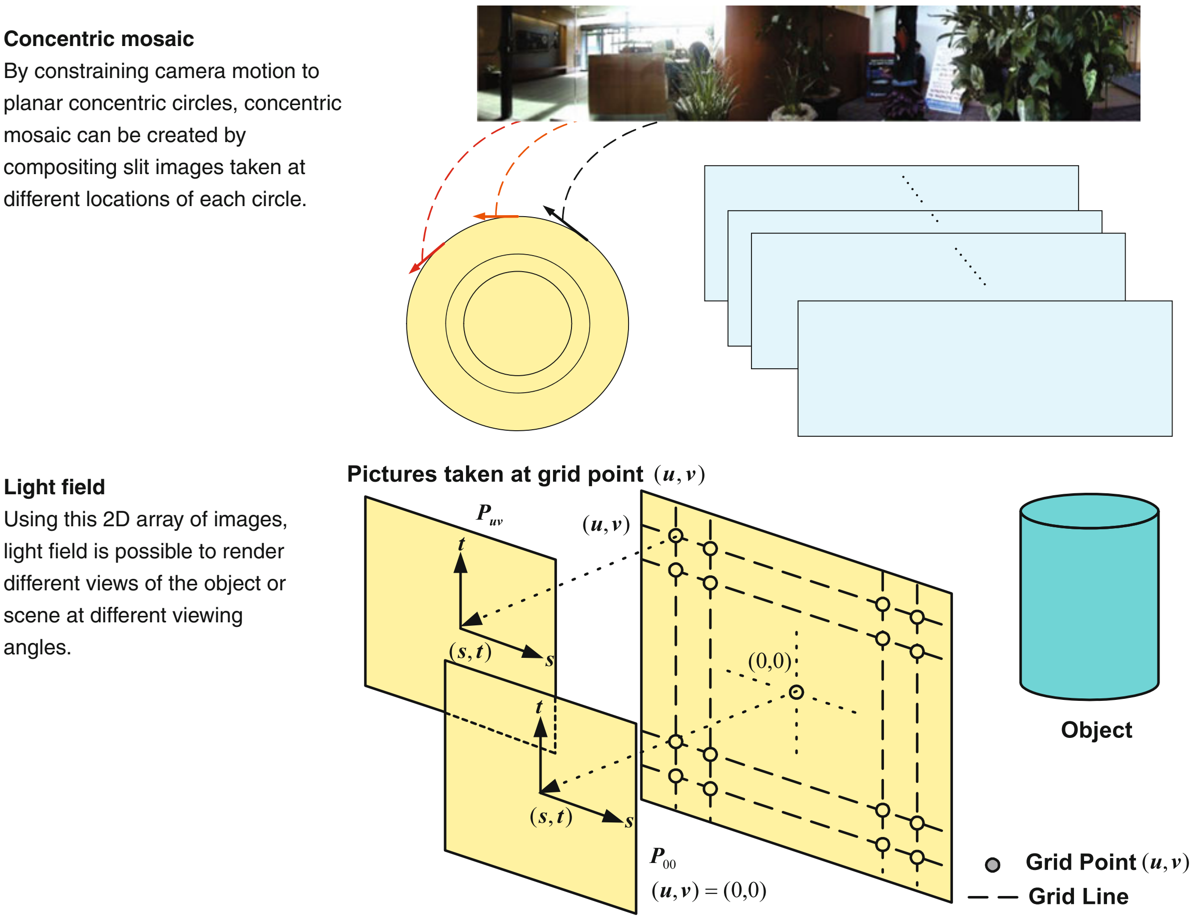
Representation

In IBR, new views of scenes are reconstructed from a collection of densely sampled images or videos. Examples include the well-known panoramas [5], light fields [9], lumigraph [8], layered depth images [13], concentric mosaics (CM) [14], etc. Figure summarizes the concept of CM and light field (see the sections on light field, lumigraph, and plenoptic function for more illustration). The reconstruction problem (i.e., rendering) is treated as a multidimensional sampling problem, where new views are generated from densely sampled images and depth maps instead of building accurate 3D model of the scenes.

Depending on the functionality required, there is a spectrum of IBR as shown in Fig. . They differ from each other in the amount of geometry information of the scenes/objects being used. At one end of the spectrum, like traditional texture mapping, very accurate geometric models of the scenes and objects say generated by animation techniques is used, but only a few images are required to generate the textures. Given the 3D models and the lighting conditions, novel views can be rendered using conventional graphic techniques. Moreover, interactive rendering with movable objects and light sources can be supported using advanced graphic hardware.

At the other extreme, light field or lumigraph rendering relies on dense sampling (by capturing more image/videos) with no or very little geometry information for rendering without recovering the exact 3D models. An important advantage of the latter is its superior image quality, compared with 3D model building for complicated real world scenes. Another important advantage is that it requires much less computational resources for rendering regardless of the scene complexity because most of the quantities involved are precomputed or recorded. This has attracted considerable attention in the computer graphic community in developing fast and efficient rendering algorithms for real-time relighting and soft-shadow generation [2, 12, 19, 22].

Broadly speaking, image-based representations can be classified according to the geometry information used into three main categories: (1) representations with no geometry, (2) representations with implicit geometry, and (3) representations with explicit geometry. 2-D panoramas, McMillan and Bishop's plenoptic modeling [11], and 3D concentric mosaics and light fields/lumigraph belong to the first category, and they can be viewed as the direct interpolation of the plenoptic function. Layere-based, object-based representations [4], pop-up light [16] using depth maps fall into the second. Finally, conventional 3D computer graphic models and other more sophisticated representations [7, 21, 22] belong to the last category. Although these representations also sample the plenoptic function, further processing of the plenoptic function has been performed to infer the scene geometry or surface property such as bidirectional reflectance distribution function (BRDF) of objects. Such image-based modeling approach has emerged as a more promising approach to enrich the photorealism and user interactivity of IBR. Moreover, since 3D models of the scenes are unavailable, conventional image-based representations are limited to the change of viewpoints and sometimes limited amount of relighting. Recently, it was found



Concentric mosaic
By constraining camera motion to planar concentric circles, concentric mosaic can be created by compositing slit images taken at different locations of each circle.

Light field
Using this 2D array of images, light field is possible to render different views of the object or scene at different viewing angles.

Image-Based Rendering, Fig. 1 Concentric mosaic and light field [3]

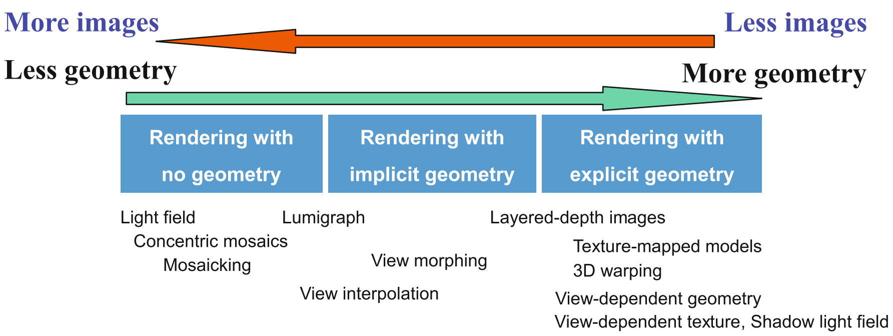


Image-Based Rendering, Fig. 2 Spectrum of IBR representations

that real-time relighting and soft-shadow computation are feasible using the IBR concepts and the associated 3D models using precomputed radiance transfer (PRT) [19] and precomputed shadow fields [22].

Earlier image-based representations are usually static, and their extension usually requires multiple camera arrays. Much research has been devoted to the capturing, compression, transmission, and processing of these dynamic representations. For a review as of 2007, see [17].

Rendering

Rendering refers to the process of rendering of new views from the images and possibly other auxiliary information captured in the representations. For early image-based representations which do not employ any geometry information, rendering can be done simply by image blending as in panoramas [5] and ray-space interpolation in light field [9]. In ray-space interpolation, each ray that corresponds to a target screen pixel is mapped to nearby sampled rays. Figure 4a shows the example renderings of a simplified light field using ray-space interpolation [17]. For more sophisticated representations which use more geometry information such as layered depth images [13], surface light field [21], and pop-up light field [16], graphics hardware has been exploited to accelerate the rendering process. The geometry information can either be implicit that relies on positional correspondences or explicit in the form of depth along known lines-of-sight or 3D coordinates. Representations of the former usually involve weakly calibrated cameras and rely on image correspondences to render new views, say by triangulating two reference images into patches according to the correspondences as in joint view triangulation (JVT) [10]. These include view interpolation, view morphing, JVT, and transfer methods with fundamental matrices and trifocal tensors. Representations employing explicit geometry include sprites, relief textures, layered depth images (LDIs), view-dependent texture, surface light field, pop-up light field, shadow light field, etc.

In general, the rendering methods can be broadly classified into three groups [17]: (1) point-based, (2) layer-based, and (3) monolithic.

Point-Based Rendering works on 3D point clouds or point correspondences, and typically each point is rendered independently. Points are mapped to the target screen through forward mapping and variants. For the 3D point X in Fig. 3, the mapping can be written as

$$X = C_r + \rho_r P_r x_r = C_t + \rho_t P_t x_t \qquad (1)$$

where x_t and x_r are homogeneous coordinates of the projection of X on target screen and reference images, respectively. C and P are camera center and projection matrix, respectively, and ρ is a scale factor. Since C_t, P_t, and the focus length f_t are known for the target view, ρ_t can be computed using the depth of X. Given x_r and ρ_r, one can compute the exact position of x_t on the target screen and transfer the color accordingly. Gaps or holes may exist due to magnification and disocclusion, and splatting techniques have been proposed to alleviate this problem. The painter's algorithm is frequently used to avoid the problem of the mapping of multiple pixels from the reference view to the same pixel in the target view.

Layered Techniques usually discretize the scene into a collection of planar layers with each layer consisting of a 3D plane with texture and optionally a transparency map. The layers can be thought of as a continuous set of polygonal models, which is amenable to conventional texture mapping and view-dependent texture mapping. Usually, each layer is rendered using either point-based or polygon meshes as in monolithic rendering techniques before being composed in the back-to-front order using the painter's algorithm to produce the final view. Layer-based rendering is also easier to implement using graphic processing unit (GPU). Since the rendering of IBR requires very low complexity, it is even possible to perform the calculation using CPU by working on individual layer or object [4].

Monolithic Rendering usually represents the geometry as continuous polygon meshes with textures, which can be readily rendered using graphics hardware. The 3D model normally consists of vertices, normals of vertices, faces, and texture mapping coordinates. The data can be stored in a variety of data formats. The most popular formats are .obj, .3ds, .max, .stl, .ply, .wrl, .dxf, etc.

Relighting, shadow generation, and interactivity have played an increasingly important role in 3D interactive rendering. The most popular algorithms are shadow mapping, shadow volume, ray tracing, precomputed radiance transfer, precomputed shadow field, etc. Some of them have better rendering quality, while others are more efficient for real-time rendering. Thanks to the development of GPU, basic lighting, and shading algorithms like shadow mapping and shadow volume have been realized on the fly. Modern GPUs can even offer programmable rendering pipelines for customized rendering effects and "shader" is a set of software instructions running on

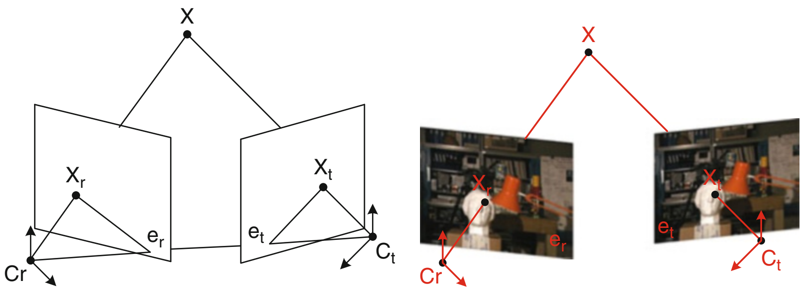


Image-Based Rendering, Fig. 3 Forward mapping

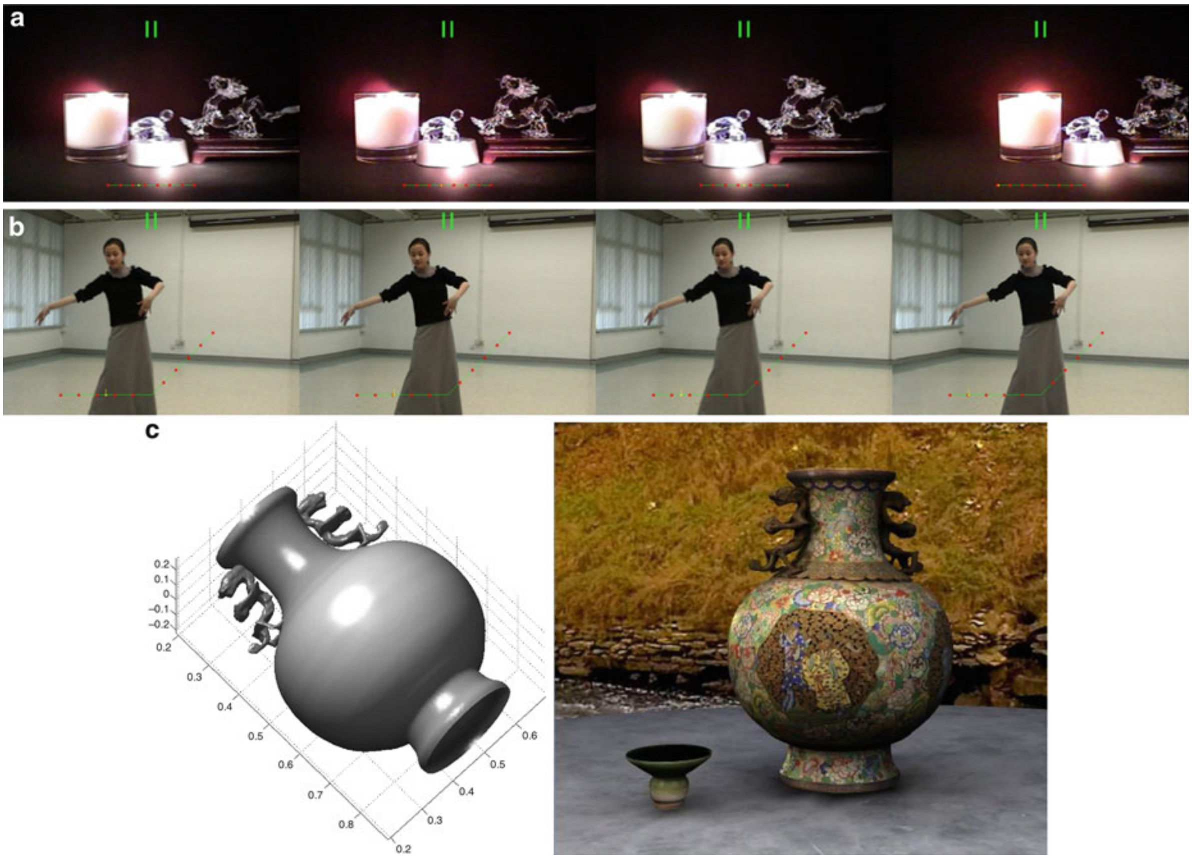


Image-Based Rendering, Fig. 4 Example renderings using (**a**) ray-space interpolation [17], (**b**) forward mapping in layered representation (with two layers – dancer and background) [4], (**c**) monolithic rendering using 3D polygonal mesh (*left*) estimated by multiview stereo and real-time rendering with shadow light field technique on GPU [23]

these GPUs to control the pipelines. Using shader programming, high-quality shadow rendering algorithms like precomputed shadow field can be done in real time. Figure 4 shows example renderings of the three techniques, and Fig. 5 summarizes the types of representations and rendering in IBR called the geometry-rendering matrix.

Compression

In general, there are two approaches to reduce the data size of image-based representations. The first one is to reduce their dimensionality, often by limiting viewpoints or scarifying some realism. Panoramas and concentric mosaics are such examples. The second approach is to exploit the high correlation

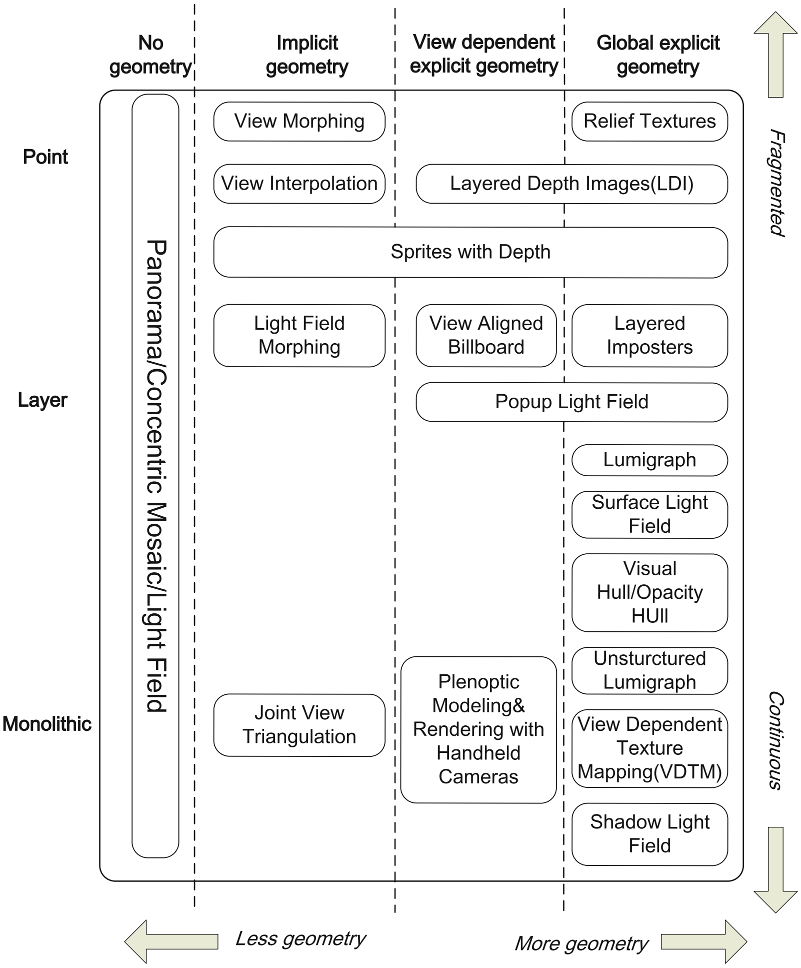


Image-Based Rendering, Fig. 5 Geometry-rendering matrix

(i.e., redundancy) within the representation using waveform coding or model-based techniques. The scene geometry may be used explicitly or implicitly. The second approach can be further classified into four broad categories: pixel-based methods, disparity compensation/prediction (DCP) methods, model-based/model-aided methods, and object-based approach.

In pixel-based methods, the correlation between adjacent image pixels is exploited using conventional techniques such as vector quantization and transform coding. In the DCP methods, scene geometry is utilized implicitly by exploiting the disparity of image pixels, resulting in better compression performance. (Disparity refers to the relative displacement of pixels in images taken at adjacent physical locations.) Model-based/model-aided approaches recover the geometry of the objects or scene in coding the observed images. The models and other information such as prediction residuals or view-dependent texture maps are then encoded. In the object-based approach, the representations are segmented into IBR objects, each with its image sequences, depth maps, and other relevant information such as shape information. The main advantage is that it helps to reduce the rendering artifacts and hence the required sampling rate. For additional references, see the section on light fields.

Unlike conventional video coding, higher dimensional image-based representations such as 3D concentric mosaics (CMs) require random access at the line level, whereas the 4D light field and lumigraph require random access at the pixel level. It is usually time-consuming to retrieve and decode a single line or pixel from the compressed which is of variable length due to entropy coding. This is referred to as the “random access problem” of IBR and is usually tackled

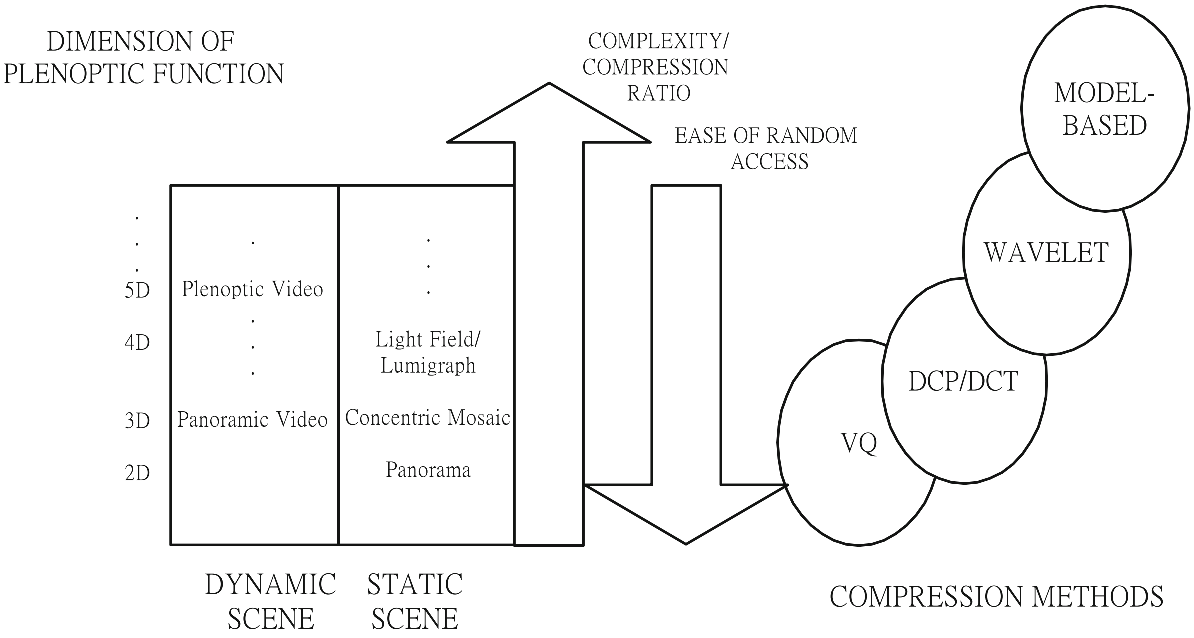


Image-Based Rendering, Fig. 6 Comparison of different image-based representation and compression methods in terms of their complexity. The ease of random access increases as the dimension of plenoptic function decreases, while the complexity and potential for compression both increase with the dimension. *DCP* disparity compensation/prediction, *VQ* vector quantization

by grouping the compressed data of several basic units for rendering (such as lines in CMs or image blocks in light fields) together and employ pointers to locate them efficiently. Moreover, interdependence of decoding resulting from DCP should be reduced to avoid decoding excessively unnecessary intermediate data. This is also required for selective transmission or decoding of the compressed representations due to their large bandwidth and storage requirement. A simple comparison of difficult image-based representations and compression methods in terms of their complexities, compression ratios, and ease of random access is shown in Fig. 6. For more information, see [15, 17] and references in the light field section.

Application

The potential for photorealistic visualization and simplicity in rendering of IBR has tremendous appeal. They have already found applications in architectural modeling [6], cultural heritage preservation [23], virtual tour, and digital museum [18], multiview TV [3, 4], etc. Other potential applications include digital edutainment, E-commerce and photorealistic modeling, and real-time rendering in computer graphics and mobile devices. Another emerging application is view synthesis in 3D and multiview videos and display.

Open Problems

Though there has been substantial progress in capturing, representing, rendering, and modeling of scenes, the ability to handle general complex scenes remains challenging for IBR. A substantial amount of work is still required to ensure robustness in handling reflection translucency, highlights, depth estimation, capturing complexity, object manipulation, etc. Since IBR uses images for rendering, interacting with IBR representations remains challenging. Recent approaches have focused on using advanced computer vision techniques, such as stereo/multiview vision and photometric stereo, and depth sensing devices to extract more geometry information from the scene so as to enhance the functionalities of IBR representations. While there has been considerable progress in relighting and interactive rendering of individual real static objects, such operations are still difficult for real and complicated scenes. For dynamic scenes, the huge amount of data and vast amount of viewpoints to be provided present one of the major challenges to IBR. Advanced algorithms for processing and manipulation of the high-dimensional representation to achieve such function as object extraction, model completion, scene inpainting, etc., are all major challenges to be addressed. Finally, the efficient transmission, compression, and display of dynamic IBR and models are also urgent issues awaiting for satisfactory solution in order for IBR to establish itself as an essential media for communication and presentation.

References

1. Adelson EH, Bergen JR (1991) The plenoptic function and the elements of early vision. In: Landy M, Movshon JA (eds) Computational models of visual processing. MIT, Cambridge, MA
2. Agrawala M, Ramamoorthi R, Heirich A, Moll L (2000) Efficient image-based methods for rendering soft shadows. In: Proceedings of ACM SIGGRAPH, New Orleans, 372–384
3. Chan SC, Shum HY, Ng KT (2007) Image-based rendering and synthesis: technological advances and challenges. IEEE Signal Process Mag 24(6):22–33
4. Chan SC, Gan ZF, Ng KT, Ho KL, Shum HY (2009) An object-based approach to image/video-based synthesis and processing for 3-D and multiview televisions. IEEE Trans. Circuits Syst Video Technol 19(6):821–831
5. Chen SE (1995) QuickTime VR – an image-based approach to virtual environment navigation. In: Proceedings of ACM SIGGRAPH, Los Angeles, 29–38
6. Debevec PE, Taylor CJ, Malik J (1996) Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In: Proceedings of ACM SIGGRAPH, New Orleans, 11–20
7. Debevec PE, Yu Y, Borshukov G (1998) Efficient view-dependent image-based rendering with projective texture-mapping. In: Eurographics workshop on rendering, Vienna, 150–116
8. Gortler SJ, Grzeszczuk R, Szeliski R, Cohen MF (1996) The lumigraph. In: Proceedings of ACM SIGGRAPH, New Orleans, 43–54
9. Levoy M, Hanrahan P (1996) Light field rendering. In: Proceedings of ACM SIGGRAPH, New Orleans, 31–42
10. Lhuillier M, Quan L (2003) Image-based rendering by joint view triangulation. IEEE Trans Circuits Syst Video Technol 13(11):1051–1063
11. McMillan L, Bishop G (1995) Plenoptic modeling: an image-based rendering system. In: Proceedings of ACM SIGGRAPH, Los Angeles, 39–46
12. Ng R, Ramamoorthi R, Hanrahan P (2004) Triple product wavelet integrals for all-frequency relighting. In: Proceedings of ACM SIGGRAPH, Los Angeles, 477–487
13. Shade J, Gortler S, He LW, Szeliski R (1998) Layered depth images. In: Proceedings of ACM SIGGRAPH, Orlando, 231–242
14. Shum HY, He LW (1999) Rendering with concentric mosaics. In: Proceedings of ACM SIGGRAPH, Los Angeles, 299–306
15. Shum HY, Kang SB, Chan SC (2003) Survey of image-based representations and compression techniques. IEEE Trans Circuits Syst Video Technol 13(11):1020–1037
16. Shum HY, Sun J, Yamazaki S, Li Y, Tang CK (2004) Pop-up light field: an interactive image-based modeling and rendering system. ACM Trans Graph 23(2):143–162
17. Shum HY, Chan SC, Kang SB (2007) Image-based rendering. Springer, New York
18. Snavely N, Simon I, Goesele M, Szeliski R, Seitz SM (2010) Scene reconstruction and visualization from community photo collections. Proc IEEE Internet Vis 98(8):1370–1390
19. Sloan P, Kautz J, Snyder J (2002) Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environment. In: Proceedings of ACM SIGGRAPH, San Antonio, 527–536
20. Szeliski R, Shum HY (1997) Creating full view panoramic image mosaics and environment maps. In: Proceedings of ACM SIGGRAPH, Los Angeles, 251–258
21. Wood DN, Azuma DI, Aldinger K, Curless B, Duchamp T, Salesin DH, Stuetzle W (2000) Surface light fields for 3D photography. In: Proceedings of ACM SIGGRAPH, New Orleans, 287–296
22. Zhou K, Hu Y, Lin S, Guo B, Shum HY (2005) Precomputed shadow fields for dynamic scenes. In: Proceedings of ACM SIGGRAPH, Los Angeles, 1196–1201
23. Zu ZY, Ng KT, Chan SC, Shum HY (2010) Image-based rendering of ancient Chinese artifacts for multi-viewdisplays – a multi-camera approach. In: Proceedings of 2010 IEEE international symposium on circuits and systems (ISCAS), Paris, 3252–3255

Image-Based Rendering (IBR)

▶Image-Based Rendering
▶Light Field
▶Plenoptic Function

Implicit Polynomial Curve

▶Algebraic Curve

Implicit Polynomial Surface

▶Algebraic Surface

Incident Light Measurement

Stephen Lin
Microsoft Research Asia, Beijing Sigma Center,
Beijing, China

Related Concepts

▸Illumination Estimation, Illuminant Estimation

Definition

Incident light measurement is the recording of incoming illumination at a given scene point or in a given scene.

Background

The distribution and intensity of light incident upon a surface point or in a scene affects the amount of reflected radiance to the camera and more generally the appearance of objects. Knowledge of the incident light can aid in shape recovery through photometric analysis techniques such as shape-from-shading and photometric stereo, or may be used in reducing appearance variation caused by lighting. Various methods have been used for measurement of incident light. Different from algorithms for illumination estimation, incident light measurement does not infer lighting from indirect scene cues such as shading, but rather obtains direct observations of the light sources.

Methods for incident light measurement typically introduce a probe or a sensor into the scene to view the incoming radiance. In general, the probes are mirrored spheres that allow for precise readings of light from a broad range of incident directions. Besides lighting distribution, the color of incident illumination may be measured using a color calibration target such as a white reference standard. Unlike illumination estimation methods, light measurement with such devices often allows for accurate recovery of both direct illumination from light sources and more subtle indirect illumination from reflected light within the scene.

Some techniques are intended to measure incident light at a certain scene point. Excluding the effects of light occluders, these methods equivalently measure far lighting that originates from distant light sources and is considered to be uniform throughout the scene. Other methods are more general in that their measurements also determine the location and brightness of near light sources, whose illumination varies within the scene. Such methods utilize triangulation, usually from two or more probes or sensors placed in the scene, to locate the positions of local light sources.

Methods

Several methods for incident light measurement are described in the following.

Spherical Probes

To measure distant illumination or the light incident at a given scene point, a common approach is to place a mirrored spherical probe at the scene point. From the reflections on the sphere, the corresponding directions of the incident light are computed from the known surface orientation of each sphere point and the mirror reflection property, which states that the incident angle of light is equal to the reflected light angle. Incident lighting environments of various scenes were measured in this manner by Debevec [1]. High dynamic range imaging was used to obtain accurate measurements of relative light source brightness.

To recover spatially variant incident lighting due to local light sources, Powell et al. [2] used three mirrored spheres at known relative positions to triangulate light source locations. For triangulation, correspondences need to be computed among the mirrored reflections of the spheres. Illumination color is also measured from the color of diffuse reflections on the spheres. Zhou and Kambhamettu [3] also employed triangulation, but instead computed correspondences in a stereo image pair of a single sphere. Shifts in specular reflections as seen from the two stereo viewpoints indicate the distance of light sources. Here, the spheres also exhibit diffuse reflection, which provides information on light intensities. Using this setup, they later proposed a method [4] based on ray tracing and convex hull computation to measure a more general light source model [5].

Hemispherical Imaging

An alternative to lighting probes is to directly place sensors within the scene. Drettakis et al. [6] employed image mosaicing of several snapshots captured within the scene to form a panoramic image of the incident lighting. Sato et al. [7] instead used a pair of omnidirectional cameras, each outfitted with a fish-eye lens. Correspondences in the omnidirectional images are computed with an omnidirectional stereo algorithm to obtain a 3D model of the incident lighting, and high dynamic range imaging is used to measure the intensity of radiance.

Color Calibration Target

The incident light color may be measured by inserting a white reference standard into the scene. Deviations from white of the reflected light indicate the color of illumination. This approach to measuring incident lighting color is described by Barnard et al. [8] for their construction of an image dataset for computational color constancy. Directional variations in illumination color may be measured by imaging the white reference standard at different orientations.

Application

Incident light measurement is often employed for augmented reality [1, 7], to ensure that inserted virtual objects exhibit an appearance consistent with the scene's illumination environment. Measurements of real-world lighting have also been utilized in computer graphics applications to give rendered objects a more natural appearance. Applications of light color measurement include evaluation of color constancy algorithms [8] and spectral reflectance recovery using multiple illumination colors [9].

References

1. Debevec P (1998) Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of ACM SIGGRAPH. ACM, New York, pp 189–198
2. Powell MW, Sarkar S, Goldgof D (2001) A simple strategy for calibrating the geometry of light sources. IEEE Trans Pattern Anal Mach Intell 23:1022–1027
3. Zhou W, Kambhamettu C (2002) Estimation of illuminant direction and intensity of multiple light sources. In: Proceedings of European conference on computer vision (ECCV). Lecture notes in computer science, vol 2353. Springer, Berlin/Heidelberg, pp 206–220
4. Zhou W, Kambhamettu C (2008) A unified framework for scene illuminant estimation. Image Vis Comput 26:415–429
5. Langer MS, Zucker SW (1997) What is a light source? In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE Computer Society, Washington, DC, pp 172–178
6. Drettakis G, Robert L, Bougnoux S (1997) Interactive common illumination for computer augmented reality. In: Proceedings of the Eurographics workshop on rendering. Springer, London, pp 45–57
7. Sato I, Sato Y, Ikeuchi K (1999) Acquiring a radiance distribution to superimpose virtual objects onto a real scene. IEEE Trans Vis Comput Graph 5:1–12
8. Barnard K, Martin L, Funt B, Coath A (2002) A data set for colour research. Color Res Appl 27:147–151
9. Han S, Sato I, Okabe T, Sato Y (2010) Fast spectral reflectance recovery using dlp projector. In: Proceedings of Asian conference on computer vision. Springer, Berlin

Inexact Matching

▶Many-to-Many Graph Matching

Information Fusion

▶Data Fusion

Inherent Optical Properties

▶Underwater Effects

Inpainting

Marcelo Bertalmío[1], Vicent Caselles[1], Simon Masnou[2] and Guillermo Sapiro[3]
[1]Universitat Pompeu Fabra, Barcelona, Spain
[2]Institut Camille Jordan, Universitè Lyon 1, Villeurbanne, France
[3]Electrical and Computer Engineering, Computer Science, and Biomedical Engineering, Duke University, Durham, NC, USA

Synonyms

Disocclusion; Error concealment; Filling in

This contribution is dedicated to the memory of Vicent Caselles, outstanding researcher, exceptional friend.

Definition

Given an image and a region Ω inside it, the inpainting problem consists in modifying the image values of the pixels in Ω so that this region does not stand out with respect to its surroundings. The purpose of inpainting might be to restore damaged portions of an image (e.g., an old photograph where folds and scratches have left image gaps) or to remove unwanted elements present in the image (e.g., a microphone appearing in a film frame). See Fig. 1. The region Ω is always given by the user, so the localization of Ω is not part of the inpainting problem. Almost all inpainting algorithms treat Ω as a hard constraint, whereas some methods allow some relaxing of the boundaries of Ω.

This definition, given for a single-image problem, extends naturally to the multi-image case; therefore, this entry covers both image and video inpainting. What is not however considered in this text is *surface* inpainting (e.g., how to fill holes in 3D scans), although this problem has been addressed in the literature.

Background

The term *inpainting* comes from art restoration, where it is also called *retouching*. Medieval artwork started to be restored as early as the Renaissance, the motives being often as much to bring medieval pictures "up to date" as to fill in any gaps. The need to retouch the image in an unobtrusive way extended naturally from paintings to photography and film. The purposes remained the same: to revert deterioration (e.g., scratches and dust spots in film) or to add or remove elements (e.g., the infamous "airbrushing" of political enemies in Stalin era USSR). In the digital domain, the inpainting problem first appeared under the name "error concealment" in telecommunications, where the need was to fill in image blocks that had been lost during data transmission. One of the first works to address automatic inpainting in a general setting dubbed it "image disocclusion" since it treated the image gap as an occluding object that had to be removed, and the image underneath would be the restoration result. Popular terms used to denote inpainting algorithms are also "image completion" and "image fill-in."

Application

The extensive literature on digital image inpainting may be roughly grouped into three categories: patch-based, sparse, and PDEs/variational methods.

From Texture Synthesis to Patch-Based Inpainting

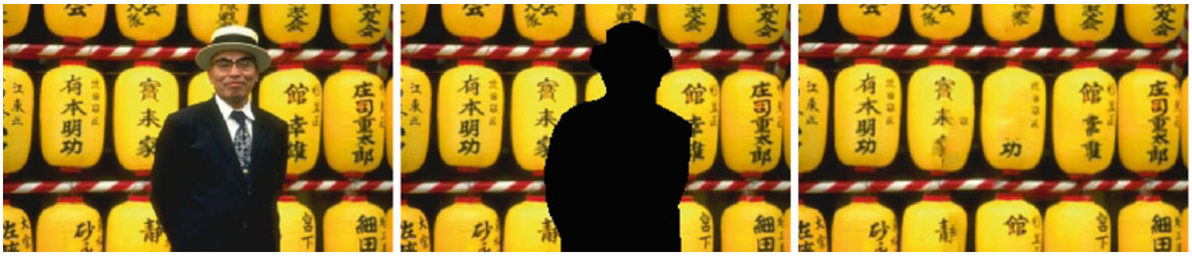
Efros and Leung [14] proposed a method that, although initially intended for texture synthesis, has proven most effective for the inpainting problem. The image gap is filled in recursively, inwards from the gap boundary: each "empty" pixel P at the boundary is filled with the value of the pixel Q (lying outside the image gap, that is, Q is a pixel with valid information) such that the neighborhood $\Psi(Q)$ of Q (a square patch centered in Q) is most similar to the (available) neighborhood $\Psi(P)$ of P. Formally, this can be expressed as an optimization problem:

$$\begin{aligned}\text{Output}(P) &= \text{Value}(Q),\; P \in \Omega,\; Q \notin \Omega,\\ Q &= \arg\min d(\Psi(P), \Psi(Q)),\end{aligned} \qquad (1)$$

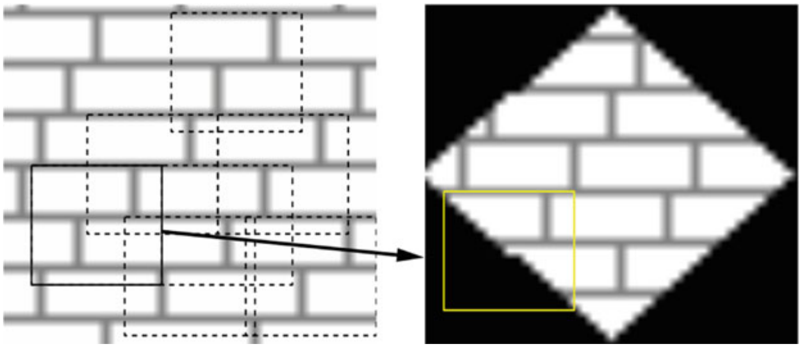
where $d(\Psi(P), \Psi(Q))$ is the sum of squared differences (SSD) among the patches $\Psi(P)$ and $\Psi(Q)$ (considering only available pixels):

$$d(\Psi_1, \Psi_2) = \sum_i \sum_j |\Psi_1(i,j) - \Psi_2(i,j)|^2, \qquad (2)$$

and the indices i, j span the extent of the patches (e.g., if Ψ is an 11×11 patch, then $0 \leq i, j \leq 10$). Once P is filled in, the algorithm marches on to the next pixel at the boundary of the gap, never going back to P (whose value is, therefore, not altered again). See Fig. 2 for an overview of the algorithm and Fig. 3 for an example of the outputs it can achieve. The results are really impressive for a wide range of images. The main shortcomings of this algorithm are its computational cost, the selection of the neighborhood size (which in the original paper is a global user-selected parameter but which should change locally, depending on image content), the filling order (which may create unconnected boundaries for some objects), and the fact that it cannot deal well with image perspective (it was intended to synthesize frontal textures; hence, neighborhoods are compared always with the same size and orientation). Also, results are poor if the image gap is very large and



Inpainting, Fig. 1 The inpainting problem. *Left*: original image. *Middle*: inpainting mask Ω, in *black*. *Right*: an inpainting result (Figure taken from [20])



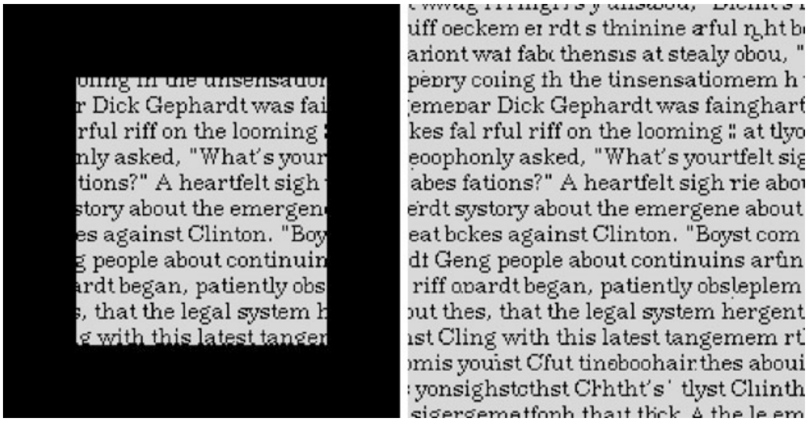
Inpainting, Fig. 2 Efros and Leung's algorithm overview (figure taken from [14]). Given a sample texture image (*left*), a new image is being synthesized one pixel at a time (*right*). To synthesize a pixel, the algorithm first finds all neighborhoods in the sample image (boxes on the *left*) that are similar to the pixels neighborhood (box on the *right*) and then randomly chooses one neighborhood and takes its *center* to be the newly synthesized pixel

disperse (e.g., an image where 80 % of the pixels have been lost due to random *salt and pepper noise*).

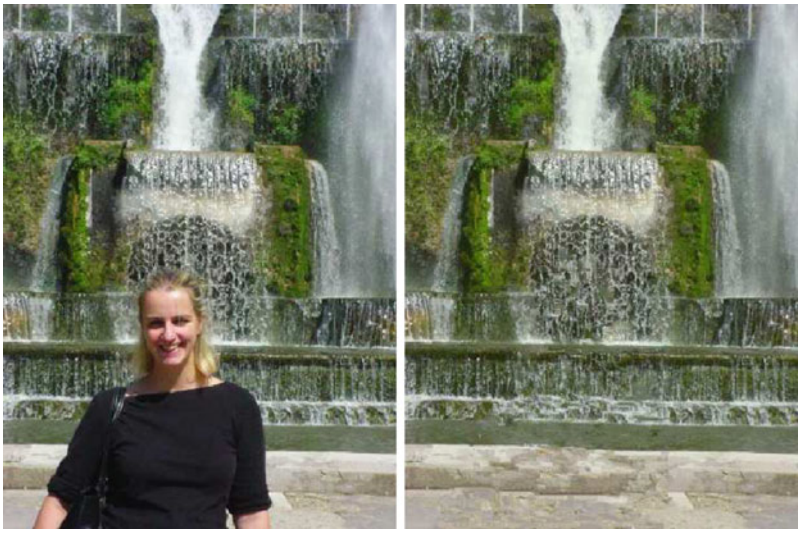
Criminisi et al. [12] improved on this work in two aspects. Firstly, they changed the filling order from the original "onion-peel" fashion to a priority scheme where empty pixels at the edge of an image object have higher priority than empty pixels on flat regions. Thus, they are able to correctly inpaint straight object boundaries which could have otherwise ended up disconnected with the original formulation. See Fig. 4. Secondly, they copy entire patches instead of single pixels, so this method is considerably faster. Several shortcomings remain, though, like the inability to deal with perspective and the need to manually select the neighborhood size (here, there are two sizes to set, one for the patch to compare with and another for the patch to copy from). Also, objects with curved boundaries may not be inpainted correctly.

Ashikhmin [2] contributed as well to improve on the original method of Efros and Leung [14]. With the idea of reducing the computational cost of the procedure, he proposed to look for the best candidate Q to copy its value to the empty pixel P not searching the whole image but only searching among the candidates of the neighbors of P which have already been inpainted. See Fig. 5. The speedup achieved with this simple technique is considerable, and also there is a very positive effect regarding the visual quality of the output. Other methods reduce the search space and computational cost involved in the candidate patch search by organizing image patches in tree structures, reducing the dimensionality of the patches with techniques like principal component analysis (PCA), or using randomized approaches.

While most image inpainting methods attempt to be fully automatic (aside from the manual setting of some parameters), there are user-assisted methods that provide remarkable results with just a little input from the user. In the work by Sun et al. [27], the user must specify curves in the unknown region, curves corresponding to relevant object boundaries. Patch synthesis is performed along these curves inside the image gap,



Inpainting, Fig. 3 *Left*: original image, inpainting mask Ω in *black*. *Right*: inpainting result obtained with Efros and Leung’s algorithm, images taken from their paper [14]



Inpainting, Fig. 4 *Left*: original image. *Right*: inpainting result obtained with the algorithm of Criminisi et al. [12], images taken from their paper

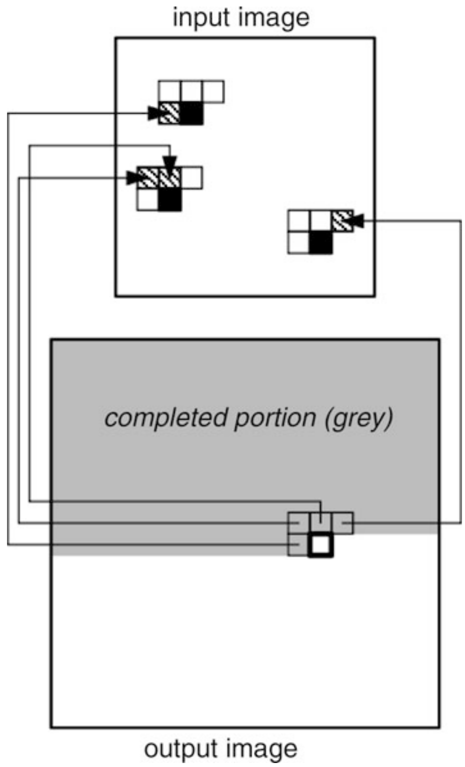
by copying from patches that lie on the segments of these curves which are outside the gap, in the “known” region. Once these curves are completed, in a process which the authors call *structure propagation*, the remaining empty pixels are inpainted using a technique like the one by Ashikhmin [2] with priorities as in Criminisi et al. [12]. Barnes et al. [5] accelerate this method and make it interactive, by employing randomized searches and combining into one step the structure propagation and texture synthesis processes of Sun et al. [27].

The Role of Sparsity

After the introduction of patch-based methods for texture synthesis by Efros and Leung [14], and image inpainting by Criminisi et al. [12], it became clear that the patches of an image provide a good dictionary to express other parts of the image. This idea has been successfully applied to other areas of image processing, for example, denoising and segmentation.

More general sparse image representations using dictionaries have proven their efficiency in the context of inpainting. For instance, using overcomplete dictionaries adapted to the representation of image geometry and texture, Elad et al. [15] proposed an image decomposition model with sparse coefficients for the geometry and texture components of the image and showed that the model can be easily adapted for image inpainting. A further description of this model follows.

Let u be an image represented as a vector in $\mathbb{R}^N$. Let the matrices D_g and D_t of sizes $N \times k_g$ and



Inpainting, Fig. 5 Ashikhmin's texture synthesis method (figure taken from [2]). Each pixel in the current L-shaped neighborhood generates a shifted candidate pixel (*black*) according to its original position (hatched) in the input texture. The best pixel is chosen among these candidates only. Several different pixels in the current neighborhood can generate the same candidate

$N \times k_t$ represent two dictionaries adapted to geometry and texture, respectively. If $\alpha_g \in \mathbb{R}^{k_g}$ and $\alpha_t \in \mathbb{R}^{k_g}$ represent the geometry and texture coefficients, then $u = D_g\alpha_g + D_t\alpha_t$ represents the image decomposition using the dictionaries collected in D_g and D_t. A sparse image representation is obtained by minimizing

$$\min_{(\alpha_g,\alpha_t):u=D_g\alpha_g+D_t\alpha_t} \|\alpha_g\|_p + \|\alpha_t\|_p, \qquad (3)$$

where $p = 0, 1$. Although the case $p = 0$ represents the sparseness measure (i.e., the number of nonzero coordinates), it leads to a nonconvex optimization problem whose minimization is more complex. The case $p = 1$ yields a convex and tractable optimization problem leading also to sparseness. Introducing the constraint by penalization (thus, in practice, relaxing it) and regularizing the geometric part of the decomposition with a total variation semi-norm penalization, Elad et al. [15] propose the variational model:

$$\min_{(\alpha_g,\alpha_t)} \|\alpha_g\|_1 + \|\alpha_t\|_1 + \lambda\|u - D_g\alpha_g - D_t\alpha_t\|_2^2 + \gamma TV(D_g\alpha_g), \qquad (4)$$

where TV denotes the total variation, $\lambda, \gamma > 0$. This model can be easily adapted to a model for image inpainting. Observe that $u - D_g\alpha_g - D_t\alpha_t$ can be interpreted as the noise component of the image and λ is a penalization parameter that depends inversely on the noise power. Then the inpainting mask can be interpreted as a region where the noise is very large (infinite). Thus, if $M = 0$ and $= 1$ identify the inpainting mask and the known part of the image, respectively, then the extension of (4) to inpainting can be written as

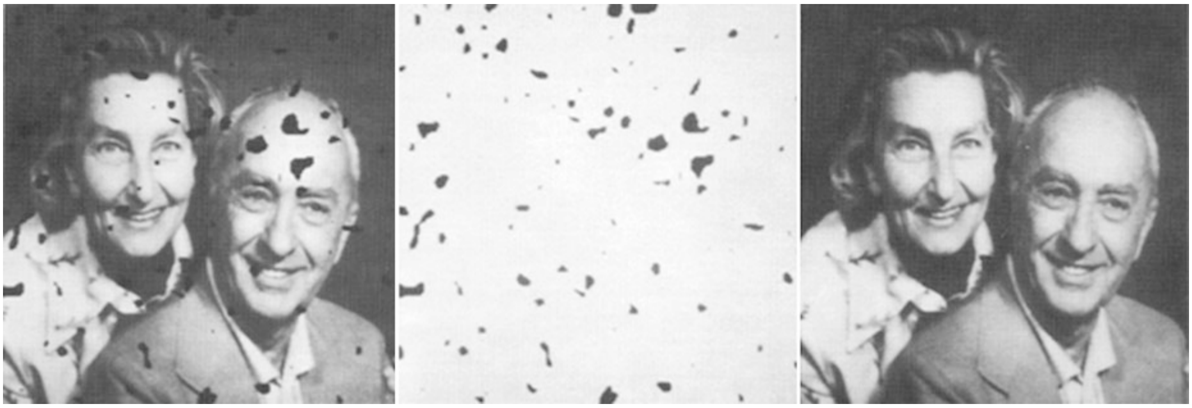
$$\min_{(\alpha_g,\alpha_t)} \|\alpha_g\|_1 + \|\alpha_t\|_1 + \lambda\|M(u - D_g\alpha_g - D_t\alpha_t)\|_2^2 + \gamma TV(D_g\alpha_g). \qquad (5)$$

Writing the energy in (5) using $u_g := D_g u$, $u_t := D_t u$ as unknown variables, it can be observed that $\alpha_g = D_g^+ u_g + r_g$, $\alpha_t = D_t^+ u_t + r_t$, where D_g^+, D_t^+ denote the corresponding pseudoinverse matrices and r_g, r_t are in the null spaces of D_g and D_t, respectively. Assuming for simplicity, as in Elad et al. [15], that $r_g = 0$, $r_t = 0$, the model (5) can be written as

$$\min_{(\alpha_g,\alpha_t)} \|D_g^+ u_g\|_1 + \|D_t^+ u_t\|_1 + \lambda\|M(u - u_g - u_t)\|_2^2 + \gamma TV(u_g). \qquad (6)$$

This simplified model is justified in Elad et al. [15] by several reasons: it is an upper bound for (5), it is easier to solve, it provides good results, it has a Bayesian interpretation, and it is equivalent to (5) if D_g and D_t are non-singular or when using the ℓ^2 norm in place of the ℓ^1 norm. The model has nice features since it permits to use adapted dictionaries for geometry and texture and treats the inpainting as missing samples, and the sparsity model is included with ℓ^1 norms that are easy to solve.

This framework has been adapted to the use of dictionaries of patches and has been extended in several directions like image denoising, filling in missing pixels (Aharon et al. [1]), color image denoising, demosaicing, and inpainting of small holes (Mairal et al. [21]



Inpainting, Fig. 6 An inpainting experiment taken from Ogden et al. [24]. The method uses a Gaussian pyramid and a series of linear interpolations, downsampling, and upsampling

and further extended to deal with multiscale dictionaries and to cover the case of video sequences in Mairal et al. [22]. To give a brief review of this model, some notation is required. Image patches are squares of size $n = \sqrt{n} \times \sqrt{n}$. Let D be a dictionary of patches represented by a matrix of size $n \times k$, where the elements of the dictionary are the columns of D. If $\alpha \in \mathbb{R}^k$ is a vector of coefficients, then $D\alpha$ represents the patch obtained by linear combination of the columns of D. Given an image $v(i,j), i,j \in \{1,\ldots,N\}$, the purpose is to find a dictionary $\hat{D}$, an image $\hat{u}$, and coefficients $\hat{\alpha} = \{\hat{\alpha}_{i,j} \in \mathbb{R}^k : i,j \in \{1,\ldots,N\}\}$ which minimize the energy

$$\min_{(\alpha,D,u)} \lambda \|v-u\|_2 + \sum_{i,j=1}^{N} \mu_{i,j} \|\alpha_{i,j}\|_0 + \sum_{i,j=1}^{N} \|D\alpha_{i,j} - R_{i,j}u\|_2, \qquad (7)$$

where $R_{i,j}u$ denotes the patch of u centered at (i,j) (dismissing boundary effects), and $\mu_{i,j}$ are positive weights. The solution of the nonconvex problem (7) is obtained using an alternate minimization: a sparse coding step where one computes $\alpha_{i,j}$ knowing the dictionary D for all i,j, a dictionary update using a sequence of one rank approximation problem to update each column of D (Aharon et al. [1]), and a final reconstruction step given by the solution of

$$\min_{u} \lambda \|v-u\|_2 + \sum_{i,j=1}^{N} \|\hat{D}\alpha_{i,j} - R_{i,j}u\|_2. \qquad (8)$$

Again, the inpainting problem can be considered as a case of nonhomogeneous noise. Defining for each pixel (i,j) a coefficient $\beta_{i,j}$ inversely proportional to the noise variance, a value of $\beta_{i,j} = 0$ may be taken for each pixel in the inpainting mask. Then the inpainting problem can be formulated as

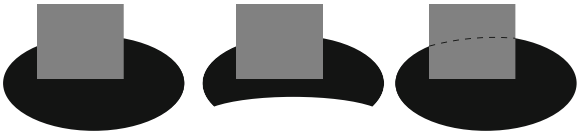
$$\min_{(\alpha,D,u)} \lambda \|\beta \otimes (v-u)\|_2 + \sum_{i,j=1}^{N} \mu_{i,j} \|\alpha_{i,j}\|_0 + \sum_{i,j=1}^{N} \|(R_{i,j}\beta) \otimes (D\alpha_{i,j} - R_{i,j}u)\|_2, \qquad (9)$$

where $\beta = (\beta_{i,j})_{i,j=1}^N$ and $\otimes$ denotes the elementwise multiplication between two vectors.

With suitable adaptations, this model has been applied to inpainting (of relatively small holes), to interpolation from sparse irregular samples and super-resolution, to image denoising, to demosaicing of color images, and to video denoising and inpainting, obtaining excellent results; see Mairal et al. [22].

PDEs and Variational Approaches

All the methods mentioned so far are based on the same principle: a missing/corrupted part of an image



Inpainting, Fig. 7 Amodal completion: the visual system automatically completes the broken edge in the *left* figure. The *middle* figure illustrates that, here, no global symmetry process is involved: in both figures, the same edge is synthesized. In such simple situation, the interpolated curve can be modeled as Euler's elastica, that is, a curve with clamped points and tangents at its extremities and with minimal oscillations

can be well synthetized by suitably sampling and copying uncorrupted patches (taken either from the image itself or built from a dictionary). A very different point of view underlies many contributions involving either a variational principle, through a minimization process, or a (non necessarily variational) partial differential equation (PDE).

An early interpolation method that applies for inpainting is due to Ogden et al. [24]. Starting from an initial image, a Gaussian filtering is built by iterated convolution and subsampling. Then, a given inpainting domain can be filled in by successive linear interpolations, downsampling, and upsampling at different levels of the Gaussian pyramid. The efficiency of such approach is illustrated in Fig. 6.

Masnou and Morel proposed in [23] to interpolate a gray-valued image by extending its isophotes (the lines of constant intensity) in the inpainting domain. This approach is very much in the spirit of early works by Kanizsa, Ullman, Horn, Mumford, and Nitzberg to model the ability of the visual system to complete edges in an occlusion or visual illusion context. This is illustrated in Fig. 7. The general completion process involves complicated phenomena that cannot be easily and univocally modeled. However, experimental results show that, in simple occlusion situations, it is reasonable to argue that the brain extrapolates broken edges using elastica-type curves, that is, curves that join two given points with prescribed tangents at these points, a total length lower than a given L, and minimize the Euler elastica energy $\int |\kappa(s)|^2 ds$, with s the curve arc length and κ the curvature.

The model by Masnou and Morel [23] generalizes this principle to the isophotes of a gray-valued image. More precisely, denoting $\tilde{\Omega}$ a domain slightly larger than Ω, it is proposed in [23] to extrapolate the isophotes of an image u, known outside Ω and valued in $[m, M]$, by a collection of curves $\{\gamma_t\}_{t\in[m,M]}$ with no mutual crossings, that coincide with the isophotes of u on $\tilde{\Omega} \setminus \Omega$ and that minimize the energy

$$\int_m^M \int_{\gamma_t} (\alpha + \beta|\kappa_{\gamma_t}|^p) ds\, dt. \tag{10}$$

Here α, β are two context-dependent parameters. This energy penalizes a generalized Euler's elastica energy, with curvature to the power $p > 1$ instead of 2, of all extrapolation curves γ_t, $t \in [m, M]$.

An inpainting algorithm, based on the minimization of (10) in the case $p = 1$, is proposed by Masnou and Morel in [23]. A globally minimal solution is computed using a dynamic programming approach that reduces the algorithmical complexity. The algorithm handles only simply connected domains, that is, those with no holes. In order to deal with color images, RGB images are turned into a luma/chrominance representation, for example, YCrCb, or Lab, and each channel is processed independently. The reconstruction process is illustrated in Fig. 8.

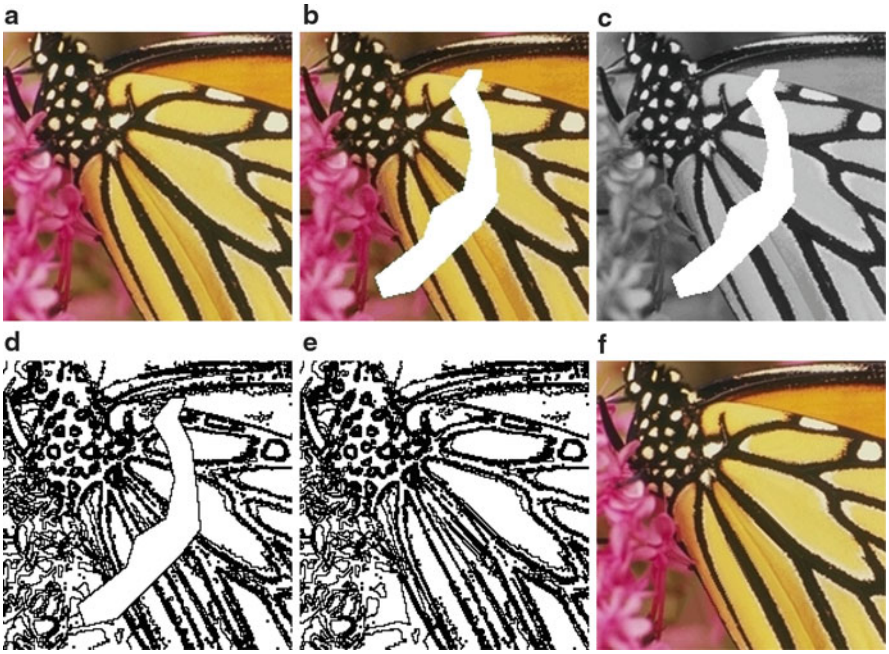
The word *inpainting*, in the image processing context, has been coined first by Bertalmío, Sapiro, Caselles, and Ballester in [6], where a PDE model is proposed in the very spirit of real paintings restoration. More precisely, u being a gray-valued image to be inpainted in Ω, a time-stepping method for the transport-like equation

$$\begin{aligned} u_t &= \nabla^{\perp} u \cdot \nabla \Delta u \quad \text{in } \Omega, \\ u &\quad \text{given in } \Omega^c \end{aligned} \tag{11}$$

is combined with anisotropic diffusion steps that are interleaved for stabilization, using the following diffusion model:

$$u_t = \varphi_\epsilon(x)\, |\nabla u|\, \nabla \cdot \frac{\nabla u}{|\nabla u|}, \tag{12}$$

where φ_ϵ is a smooth cutoff function that forces the equation to act only in Ω, and $\nabla \cdot (\nabla u / |\nabla u|)$ is the curvature along isophotes. This diffusion equation, which has been widely used for denoising an image while preserving its edges, compensates any shock possibly created by the transport-like equation. What is the meaning of Eq. (11)? Following Bertalmío et al. [6], Δu is a measure of image smoothness, and stationary points for the equation are images for which Δu



Inpainting, Fig. 8 (**a**) is the original image and (**b**) the image with occlusions in *white*. The luminance channel is shown in figure (**c**). A few isophotes are drawn in figure (**d**), and their reconstruction by the algorithm of Masnou and Morel [23] is given in figure (**e**). Applying the same method to the luminance, hue, and saturation channels yields the final result of figure (**f**)

is constant along the isophotes induced by the vector field $\nabla^{\perp}u$. Equation (11) is not explicitly a transport equation for Δu, but, in the equivalent form,

$$u_t = -\nabla^{\perp}\Delta u \cdot \nabla u, \tag{13}$$

it is a transport equation for u being convected by the field $\nabla^{\perp}\Delta u$. Following Bornemann and März [9], this field is in the direction of the level lines of Δu, which are related to the Marr-Hildreth edges. Indeed, the zero crossings of (a convoluted version of) Δu are the classical characterization of edges in the celebrated model of Marr and Hildreth. In other words, as in the real paintings restoration, the approach of Bertalmío et al. [6] consists in conveying the image intensities along the direction of the edges, from the boundary of the inpainting domain Ω toward the interior. The efficiency of such approach is illustrated in Fig. 9. From a numerical viewpoint, the transport equation and the anisotropic diffusion can be implemented with classical finite difference schemes. For color images, the coupled system can be applied independently to each channel of any classical luma/chrominance representation. There is no restriction on the topology of the inpainting domain.

Another perspective on this model is provided by Bertalmío, Bertozzi, and Sapiro in [7], where connections with the classical Navier-Stokes equation of fluid dynamics are shown. Indeed, the steady-state equation of Bertalmío et al. [6],

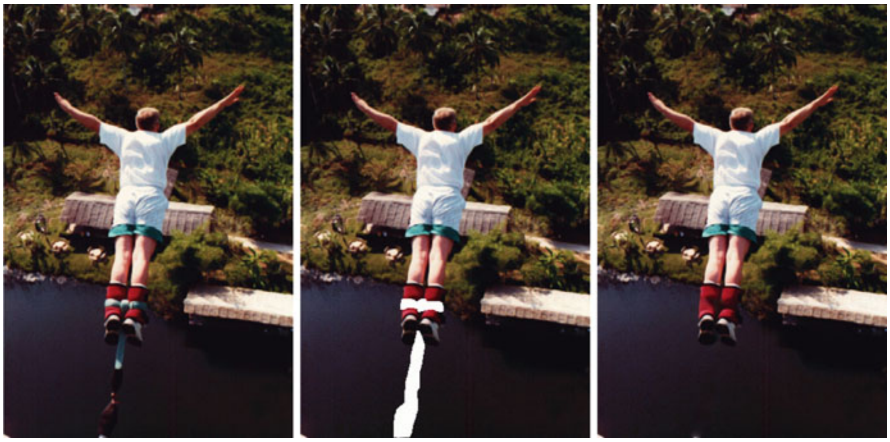
$$\nabla^{\perp}u \cdot \nabla\Delta u = 0,$$

is exactly the equation satisfied by steady-state inviscid flows in the two-dimensional incompressible Navier-Stokes model. Although the anisotropic diffusion equation (12) is not the exact counterpart of the viscous diffusion term used in the Navier-Stokes model for incompressible and Newtonian flows, a lot of the numerical knowledge on fluid mechanics seems to be adaptable to design stable and efficient schemes for inpainting. Results in this direction are shown in Bertalmío et al. [7].

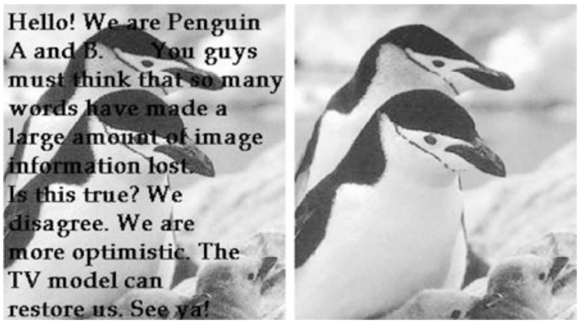
Chan and Shen propose in [10] a denoising/inpainting first-order model based on the joint minimization of a quadratic fidelity term outside Ω and a total variation criterion in Ω, that is, the joint energy

$$\int_A |\nabla u| dx + \frac{\lambda}{2}\int_{\Omega} |u - u_0|^2 dx,$$

with $A \Supset \Omega$ the image domain and λ a Lagrange multiplier. The existence of solutions to this problem follows easily from the properties of functions of bounded variation. As for the implementation, Chan and Shen look for critical points of the energy



Inpainting, Fig. 9 An experiment taken from Bertalmío et al. [6]. *Left*: original image. *Middle*: a user-defined mask. *Right*: the result with the algorithm of [6]



Inpainting, Fig. 10 An experiment taken from Chan and Shen [10]. *Left*: original image. *Right*: after denoising and removal of text

using a Gauss-Jacobi iteration scheme for the linear system associated to an approximation of the Euler-Lagrange equation by finite differences. More recent approaches to the minimization of total variation with subpixel accuracy should nowadays be preferred. From the phenomenological point of view, the model of Chan and Shen [10] yields inpainting candidates with the smallest possible isophotes. It is therefore more suitable for thin or sparse domains. An illustration of the model's performances is given in Fig. 10.

Turning back to the criterion (10), a similar penalization on $\tilde{\Omega}$ of both the length and the curvature of all isophotes of an image u yields two equivalent forms, in the case where u is smooth enough (see Masnou and Morel [23]):

$$\int_{-\infty}^{+\infty}\int_{\{u=t\}\cap\tilde{\Omega}}(\alpha+\beta|\kappa|^{p})\,ds\,dt = \int_{\tilde{\Omega}}|\nabla u|\left(\alpha+\beta\left|\nabla\cdot\frac{\nabla u}{|\nabla u|}\right|^{p}\right)dx. \qquad (14)$$

There have been various contributions to the numerical approximation of critical points for this criterion. A fourth-order time-stepping method is proposed by Chan et al. in [11] based on the approximation of the Euler-Lagrange equation, for the case $p = 2$, using upwind finite differences and a min-mod formula for estimating the curvature. Such high-order evolution method suffers from well-known stability and convergence issues that are difficult to handle.

A model, slightly different from (14), is tackled by Ballester et al. in [4] using a relaxation approach. The key idea is to replace the second-order term $\nabla\cdot\frac{\nabla u}{|\nabla u|}$ with a first-order term, depending on an auxiliary variable. More precisely, Ballester et al. study in [4] the minimization of

$$\int_{\tilde{\Omega}}|\nabla\cdot\theta|^{p}(a+b|\nabla G*u|)dx+\alpha\int_{\tilde{\Omega}}(|\nabla u|-\theta\cdot\nabla u)dx,$$

under the constraint that θ is a vector field with subunit modulus and prescribed normal component on the boundary of $\tilde{\Omega}$, and u takes values in the same range as in Ω^{c}. Clearly, θ plays the role of $\nabla u/|\nabla u|$, but the new criterion is much less singular. As for G, it is a regularizing kernel introduced for technical reasons in order to ensure the existence of a minimizing couple (u,θ).

The main difference between the new relaxed criterion and (14), besides singularity, is the term $\int_{\tilde{\Omega}} |\nabla \cdot \theta|^p$ which is more restrictive, despite the relaxation, than $\int_{\tilde{\Omega}} |\nabla u| \left| \nabla \cdot \frac{\nabla u}{|\nabla u|} \right|^p dx$. However, the new model has a nice property: a gradient descent with respect to (u, θ) can be easily computed and yields two coupled second-order equations whose numerical approximation is standard. Results obtained with this model are shown in Fig. 11.

The Mumford-Shah-Euler model by Esedoglu and Shen [16] is also variational. It combines the celebrated Mumford-Shah segmentation model for images and the Euler's elastica model for curves. Being u_0 the original image defined on a domain A, and $\Omega \subset A$ the inpainting domain, Esedoglu and Shen propose to find a piecewise weakly smooth function u, that is a function with integrable squared gradient out of a discontinuity set $K \subset A$, that minimizes the criterion

$$\int_{A\setminus\Omega} \lambda \|u - u_0\|^2 dx + \int_{A\setminus K} \gamma |\nabla u|^2 dx + \int_K (\alpha + \beta\,\kappa^2) ds,$$

where $\alpha, \beta, \gamma, \lambda$ are positive parameters. The resulting image is not only reconstructed in the inpainting domain $\mathit{\Omega}$, but also segmented all over A since the original image is not imposed as a hard constraint.

Two numerical approaches to the minimization of this model are discussed in Esedoglu and Shen [16]: first, a level set approach based on the representation of K as the zero-level set of a sequence of smooth functions that concentrate, and the explicit derivation, using finite differences, of the Euler-Lagrange equations associated with the criterion; second, a Γ-convergence approach based on a result originally conjectured by De Giorgi and recently proved by Röger and Schätzle in dimensions 2,3. In both cases, the final system of discrete equations is of order four, facing again difficult issues of convergence and stability.

More recently, following the work of Grzibovskis and Heintz on the Willmore flow, Esedoglu et al. [17] have addressed the numerical flow associated with the Mumford-Shah-Euler model using a promising convolution/thresholding method that is much easier to handle than the previous approaches.

Tschumperlé proposes in [28] an efficient second-order anisotropic diffusion model for multivalued image regularization and inpainting. Given a $\mathbb{R}^N$-valued image u known outside Ω, and starting from an initial rough inpainting obtained by straightforward advection of boundary values, the pixels in the inpainting domain are iteratively updated according to a finite difference approximation to the equations

$$\frac{\partial u_i}{\partial t} = \operatorname{trace}(T\,\nabla^2 u_i), \qquad i \in \{1, \cdots, N\}.$$

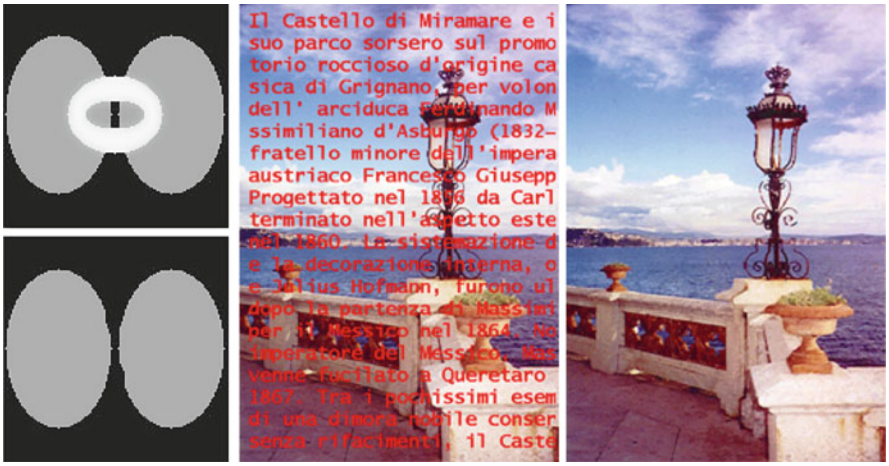
Here, T is the tensor field defined as

$$T = \frac{1}{(1 + \lambda_{\min} + \lambda_{\max})^{\alpha_1}} v_{\min} \otimes v_{\min} + \frac{1}{(1 + \lambda_{\min} + \lambda_{\max})^{\alpha_2}} v_{\max} \otimes v_{\max},$$

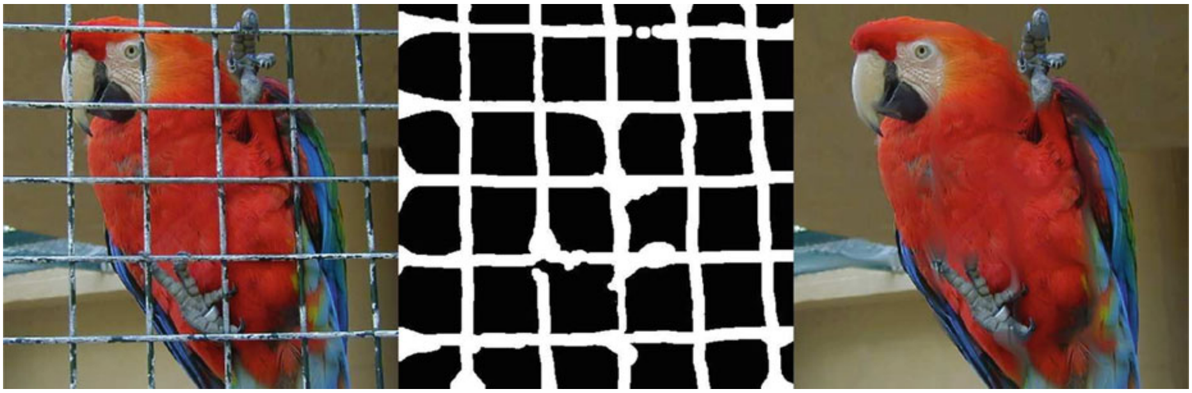
with $0 < \alpha_1 \ll \alpha_2$, and $\lambda_{\min}$, $\lambda_{\max}$, $v_{\min}$, $v_{\max}$ are the eigenvalues and eigenvectors, respectively, of $G_\sigma * \sum_{i=1}^N \nabla u_i \otimes \nabla u_i$, being G_σ a smoothing kernel and $\sum_{i=1}^N \nabla u_i \otimes \nabla u_i$ the classical structure tensor, which is known for representing well the local geometry of u. Figure 12 reproduces an experiment taken from Tschumperlé [28].

The approach of Auroux and Masmoudi in [3] uses the PDE techniques that have been developed for the inverse conductivity problem in the context of crack detection. The link with inpainting is the following: missing edges are modeled as cracks, and the image is assumed to be smooth out of these cracks. Given a crack, two inpainting candidates can be obtained as the solutions of the Laplace equation with Neumann condition along the crack and either a Dirichlet or a Neumann condition on the domain's boundary. The optimal cracks are those for which the two candidates are the most similar in quadratic norm, and they can be found through topological analysis, that is, they correspond to the set of points where putting a crack mostly decreases the quadratic difference. Both the localization of the cracks and the associated piecewise smooth inpainting solutions can be found using fast and simple finite difference schemes.

Finally, Bornemann and März propose in [9] a first-order model to advect the image information along the integral curves of a coherence vector field that extends in Ω the dominant directions of the image gradient. This coherence field is explicitly defined, at every point, as the normalized eigenvector to the minimal eigenvalue of a smoothed structure tensor whose computation carefully avoids boundary biases in the vicinity of $\partial\Omega$. Denoting c the coherence field,


Il Castello di Miramare e i
suo parco sorsero sul promo
torio roccioso d'origine ca
sica di Grignano, per volon
dell' arciduca Ferdinando M
ssimiliano d'Asburgo (1832-
fratello minore dell'impera
austriaco Francesco Giusepp
Progettato nel 1856 da Carl
terminato nell'aspetto este
nel 1860. La sistemazione d
e la decorazione interna, o
e Julius Hofmann, furono ul
dopo la partenza di Massimi
per il Messico nel 1864. No
imperatore del Messico, Mas
venne fucilato a Queretaro
1867. Tra i pochissimi esem
di una dimora nobile conser
senza rifacimenti, il Caste

Inpainting, Fig. 11 Two inpainting results obtained with the model proposed by Ballester et al. [4]. Observe in particular how curved edges are restored



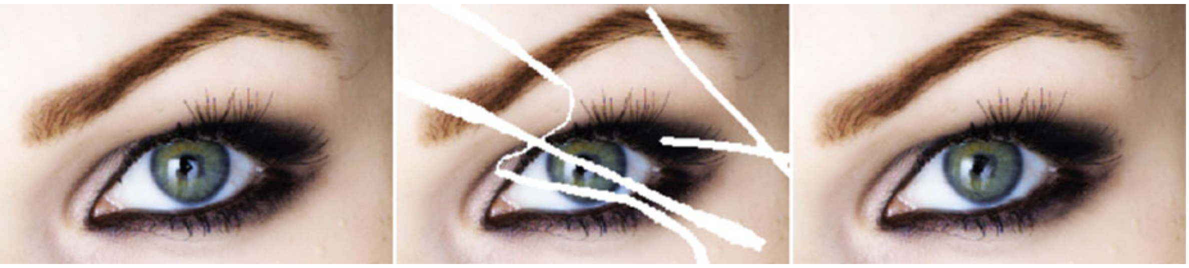
Inpainting, Fig. 12 An inpainting experiment (the *middle* image is the mask defined by the user) taken from Tschumperlé [28]

Bornemann and März show that the equation $c \cdot \nabla u = 0$ with Dirichlet boundary constraint can be obtained as the vanishing viscosity limit of an efficient fast-marching scheme: the pixels in Ω are synthesized one at a time, according to their distance to the boundary. The new value at a pixel p is a linear combination of both known and previously generated values in a neighborhood of p. The key ingredient of the method is the explicit definition of the linear weights according to the coherence field c. Although the Bornemann-März model requires a careful tune of four parameters, it is much faster than the PDE approaches mentioned so far and performs very well, as illustrated in Fig. 13.

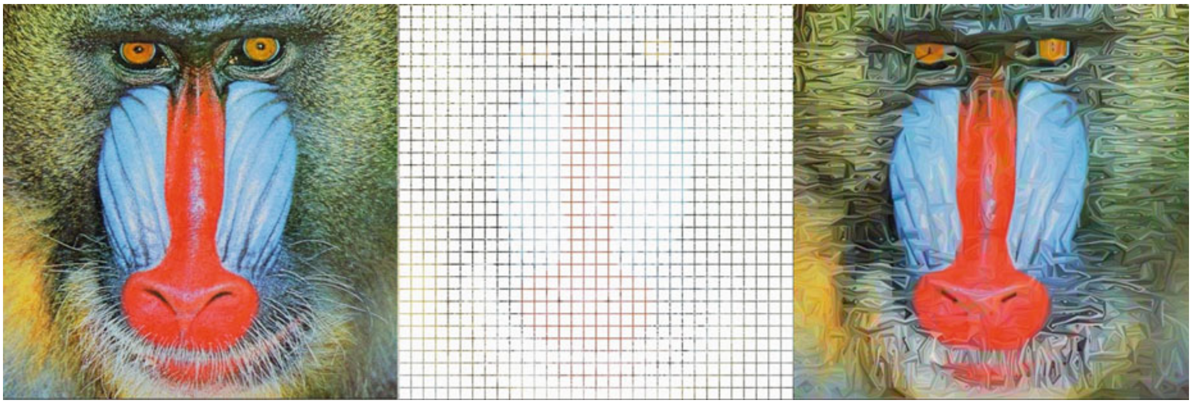
Combining and Extending PDEs and Patch Models

In general, most PDE/variational methods that have been presented so far perform well for inpainting either thin or sparsely distributed domains. However, there is a common drawback to all these methods: they are unable to restore texture properly, and this is particularly visible on large inpainting domains, for instance, in the inpainting result of Fig. 12 where the diffusion method is not able to recover the parrot's texture. On the other hand, patch-based methods are not able to handle sparse inpainting domains like in Fig. 14, where no valid squared patch can be found that does not reduce to a point. On the contrary, most PDE/variational methods remain applicable in such situation, like in Fig. 14 where the model proposed by Masnou and Morel [23] yields the inpainting result. Obviously, some geometric information can be recovered, but no texture.

There have been several attempts to explicitly combine PDEs and patch-based methods in order to handle properly both texture and geometric structures. The contribution of Criminisi et al. [12] was mentioned



Inpainting, Fig. 13 An inpainting experiment taken from Bornemann and März [9], with a reported computation time of 0.4 s

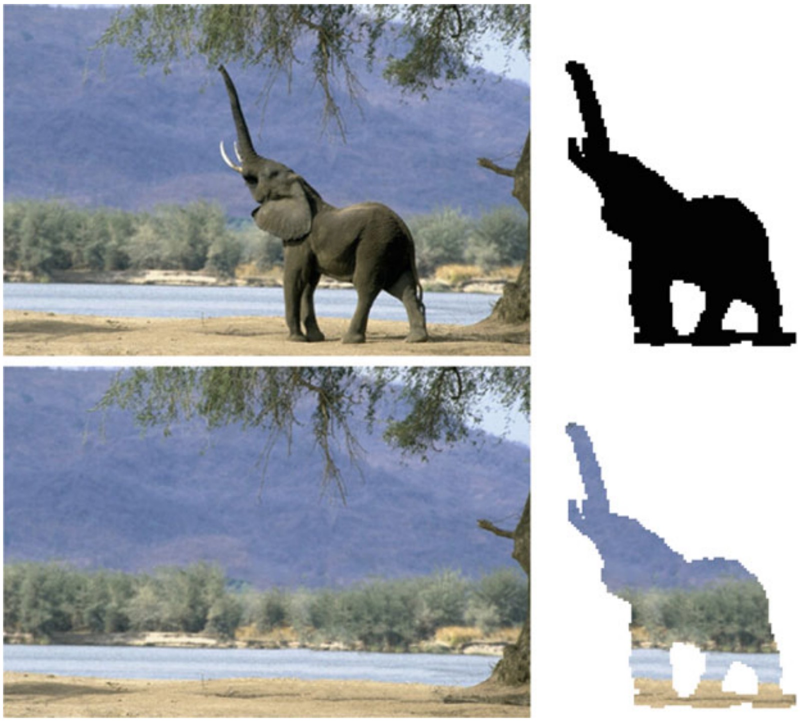


Inpainting, Fig. 14 A picture of a mandrill, the same picture after removal of 15 × 15 squares (more than 87 % of the pixels are removed), and the reconstruction with the method introduced by Masnou and Morel [23] using only the one-pixel-wide information at the squares' boundaries

already. The work of Bertalmío et al. [8] uses an additive decomposition of the image to be inpainted into a geometric component that contains all edges information, and a texture component. Then the texture image is restored using the Efros and Leung's algorithm of [14], while the geometric image is inpainted following the method proposed in Bertalmío et al. [6] (several subsequent works have proposed other methods for the individual reconstruction of each component). The final image is obtained by addition of the restored texture and geometric components. In a few situations where the additive decomposition makes sense, this approach does indeed improve the result and extends the applications domain of inpainting.

In Komodakis and Tziritas [20], the authors combine variational and patch-based strategies by defining an inpainting energy over a graph whose nodes are the centers of patches over the image. The inpainting energy has two terms, one being a texture synthesis term and the other measuring the similarity of the overlapping area of two neighboring patches (centered on nodes which are neighbors in the graph). By minimizing this energy with belief propagation, a label is assigned to each node, which amounts to copying the patch corresponding to the label over the position of the node. The results are very good on a variety of different images (e.g., Fig. 1), and the method is fast. Some potential issues are the following: there is no assurance that the iterative process converges to a global minimum, and visual artifacts may appear since the method uses a fixed grid and entire patches are copied for each pixel of the mask.

The work by Drori et al. in [13] does not involve any explicit geometry/texture decomposition, but the search for similar neighborhoods is guided by a prior rough estimate of the inpainted values using a multiscale sampling and convolution strategy, in the very spirit of Ogden et al. [24]. In addition, in contrast with many patch-based methods, the dictionary of valid patches is enriched using rotations, rescalings, and reflections. An example extracted from Drori et al. [13] is shown in Fig. 15.



Inpainting, Fig. 15 An experiment from Drori et al. [13] illustrating the proposed multiscale diffusion/patch-based inpainting method. The *upper-left* image is the original, the *upper-right* image contains the mask defined by the user, the *bottom-left* image is the result, and the *bottom-right* image shows what has been synthesized in place of the elephant

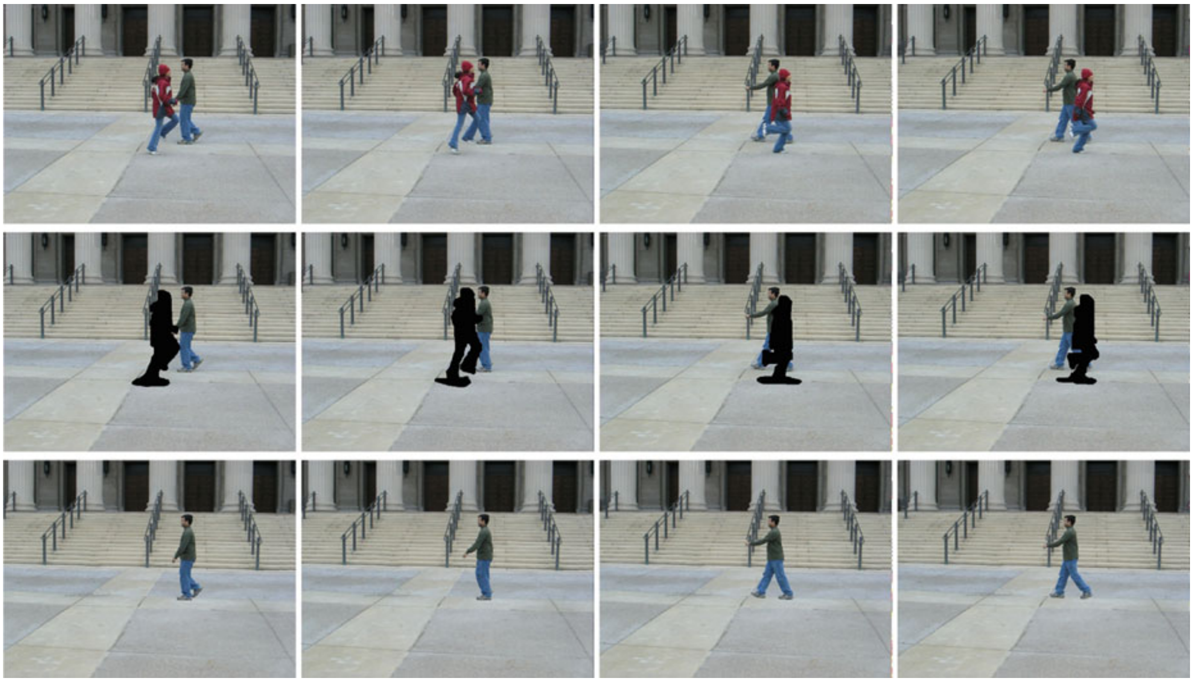
Beyond Single-Image Inpainting

All the methods mentioned above involve just a single image. For the multi-image case, there are two possible scenarios: video inpainting and inpainting a single image using information from several images.

Basic methods for video inpainting for data transmission (where the problem is known as "error concealment" and involves restoring missing image blocks) and for film restoration applications (dealing with image gaps produced by dust, scratches, or the abrasion of the material) assume that the missing data changes location in correlative frames and therefore use motion estimation to copy information along pixel trajectories. A particular difficulty in video inpainting for film restoration is that, for good visual quality of the outputs, the detection of the gap and its filling in are to be tackled jointly and in a way which is robust to noise, usually employing probabilistic models in a Bayesian framework; see, for example, the book by Kokaram [19].

Wexler et al. [29] propose a video inpainting algorithm that extends to space-time the technique of Efros and Leung [14] and combines it with the idea of coherence among neighbors developed by Ashikhmin [2]. First, for each empty pixel P, they consider a space-time cube centered in P, compare it with all possible cubes in the video, find the most similar, and keep its center pixel Q, which will be the correspondent of P. For each cube the information considered and compared is not only color but also motion vectors. Then, instead of copying the value of Q to P, they copy to P the average of all the values of the shifted correspondents of the neighbors of P: for instance, if R is at the *right* of P, and S is the correspondent of R, then the pixel to the *left* of S will be involved in the average to fill in P. This is based on the idea by Ashikhmin [2], see Fig. 5. The shortcomings of this video inpainting method are that the results present significant blur (due to the averaging), it seems to be limited only to static-camera scenarios (probably due to the simple motion estimation procedure involved) and periodic motion without change of scale, and the computational cost is quite high (due to the comparison of 3D blocks).

Shiratori et al. [26] perform video inpainting by firstly inpainting the motion field with a patch-based technique like that of Efros and Leung [14] and then propagating the colors along the (inpainted) motion trajectories. The method assumes that motion information is sufficient to fill in holes in videos, which is not always the case (e.g., with a static hole over a



Inpainting, Fig. 16 *Top row*: some frames from a video. *Middle row*: inpainting mask Ω in black. *Bottom row*: video inpainting results obtained with the algorithm of Patwardhan et al. [25]

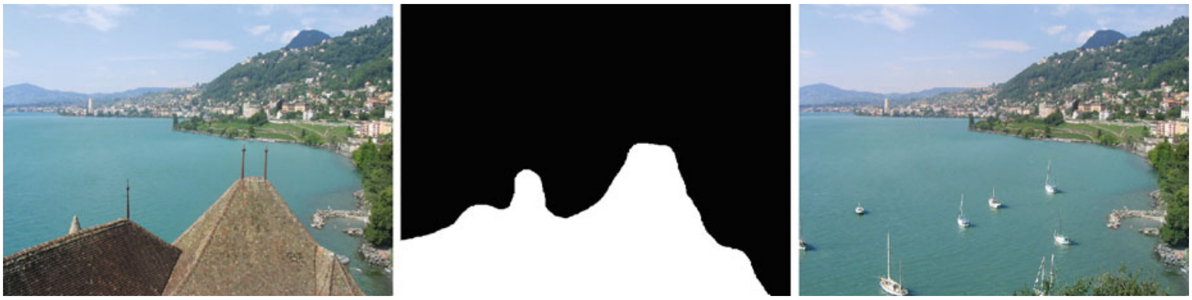
static region). The results present some blurring, due to the bilinear interpolation in the color propagation step.

Patwardhan et al. [25] propose a video inpainting method consisting of three steps. In the first step they decompose the video sequence into binary motion layers (foreground and background), which are used to build three image *mosaics* (a mosaic is the equivalent of a panorama image created by stitching together several images): one mosaic for the foreground, another for the background, and a third for the motion information. The other two steps of the algorithm perform inpainting, first from the foreground and then from the background: these inpainting processes are aided and sped up by using the mosaics computed in the first step. See Fig. 16 for some results. The algorithm is limited to sequences where the camera motion is approximately parallel to the image plane and foreground objects move in a repetitive fashion and do not change size: these restrictions are imposed so that a patch-synthesis algorithm like that of Efros and Leung [14] can be used.

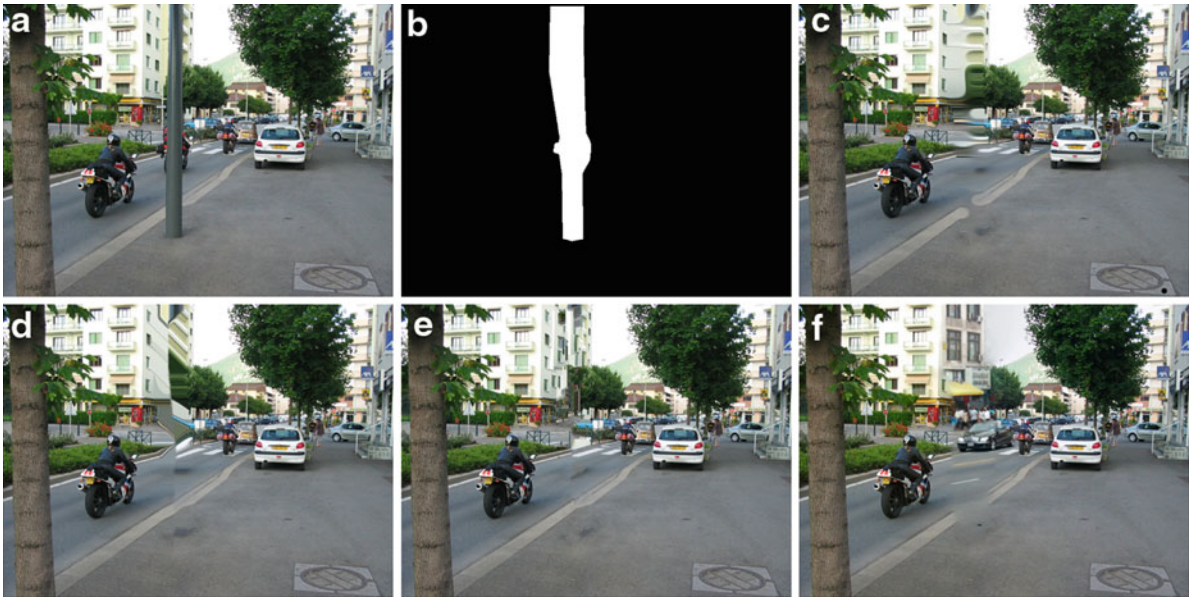
Hays and Efros [18] perform inpainting of a single image using information from a database with several millions of photographs. They use a scene descriptor to reduce the search space from two million to two hundred images, those images from the database which are *semantically* closer to the image the user wants to inpaint. Using template matching, they align the 200 best matching scenes to the local image around the region to inpaint. Then they composite each matching scene into the target image using seam finding and image blending. Several outputs are generated so the user may select among them, and the results can be outstanding; see Fig. 17. The main shortcoming of this method is that it relies on managing and operating a huge image database. When the algorithm fails, it can be due to a lack of good scene matches (if the target image is atypical), or because of *semantic violations* (e.g., failure to recognize people, hence copying only part of them), or in the case of uniformly textured backgrounds (where this algorithm might not find the precise same texture in another picture of the database).

Open Problems

Inpainting is a very challenging problem, and it is far from being solved; see Fig. 18. Patch-based methods



Inpainting, Fig. 17 *Left*: original image. *Middle*: inpainting mask Ω, in white. *Right*: inpainting result obtained with the method by Hays and Efros [18], images taken from their paper



Inpainting, Fig. 18 An example where no inpainting method seems to work. (**a**) Original image, from the database provided by Hays and Efros [18]. (**b**) In *white*, the mask to be inpainted, which is not the initial mask proposed by Hayes and Efros but derives from the fuzzy mask actually used by their algorithm. (**c**) Result courtesy of D. Tschumperlé using the algorithm from [28]. (**d**) Result courtesy of T. März and F. Bornemann using the algorithm from [9]. (**e**) Result using a variant of the algorithm from Criminisi et al. [12]. (**f**) Result from Hays and Efros [18]

work best in general, although for some applications (e.g., very spread, sparsely distributed gap Ω) geometry-based methods might be better suited. And when the image gap lies on a singular location, with surroundings that cannot be found anywhere else, then all patch-based methods give poor results, regardless if they consider or not geometry. For video inpainting the situation is worse; the existing algorithms are few and with very constraining limitations on camera and object motion. Because video inpainting is very relevant in cinema postproduction, in order to replace the current typical labor intensive systems, important developments are expected in the near future.

References

1. Aharon M, Elad M, Bruckstein A (2006) K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans Signal Process 54(11):4311
2. Ashikhmin M (2001) Synthesizing natural textures. In: Proceedings of the ACM symposium on interactive 3D graphics, Chapel Hill. ACM, pp 217–226

3. Auroux D, Masmoudi M (2006) A one-shot inpainting algorithm based on the topological asymptotic analysis. Comput Appl Math 25:1–17
4. Ballester C, Bertalmío, M, Caselles V, Sapiro G, Verdera J (2001) Filling-in by joint interpolation of vector fields and gray levels. IEEE Trans Image Process 10(8):1200–1211
5. Barnes C, Shechtman E, Finkelstein A, Goldman DB (2009) Patchmatch: a randomized correspondence algorithm for structural image editing. ACM Trans Graph 28(3):2
6. Bertalmío M, Sapiro G, Caselles V, Ballester C (2000) Image inpainting. In: Proceedings of SIGGRAPH'00, New Orleans, USA, pp 417–424
7. Bertalmío M, Bertozzi A, Sapiro G (2001) Navier-Stokes, fluid dynamics, and image and video inpainting. In: Proceedings of the IEEE international conference on computer vision and pattern recognition (CVPR), Hawaï
8. Bertalmío M, Vese L, Sapiro G, Osher S (2003) Simultaneous structure and texture image inpainting. IEEE Trans Image Process 12(8):882–889
9. Bornemann F, März T (2007) Fast image inpainting based on coherence transport. J Math Imaging Vis 28(3): 259–278
10. Chan TF, Shen J (2001) Mathematical models for local deterministic inpaintings. SIAM J Appl Math 62(3): 1019–1043
11. Chan TF, Kang SH, Shen J (2002) Euler's elastica and curvature based inpainting. SIAM J Appl Math 63(2): 564–592
12. Criminisi A, Pérez P, Toyama K (2004) Region filling and object removal by exemplar-based inpainting. IEEE Trans Image Process 13(9):1200–1212
13. Drori I, Cohen-Or D, Yeshurun H (2003) Fragment-based image completion. In: Proceedings of SIGGRAPH'03, vol 22(3), pp 303–312
14. Efros AA, Leung TK (1999) Texture synthesis by non-parametric sampling. In: Proceedings of the international conference on computer vision, Kerkyra, vol 2, pp 1033
15. Elad M, Starck JL, Querre P, Donoho DL (2005) Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). Appl Comput Harmon Anal 19(3):340–358
16. Esedoglu S, Shen J (2002) Digital image inpainting by the Mumford-Shah-Euler image model. Eur J Appl Math 13:353–370
17. Esedoglu S, Ruuth S, Tsai R (2008) Threshold dynamics for high order geometric motions. Interfaces Free Boundaries 10(3):263–282
18. Hays J, Efros AA (2008) Scene completion using millions of photographs. Commun ACM 51(10):87–94
19. Kokaram AC (1998) Motion picture restoration: digital algorithms for artefact suppression in degraded motion picture film and video. Springer, London
20. Komodakis N, Tziritas G (2007) Image completion using efficient belief propagation via priority scheduling and dynamic pruning. IEEE Trans Image Process 16(11):2649
21. Mairal J, Elad M, Sapiro G (2008) Sparse representation for color image restoration. IEEE Trans Image Process 17(1):53
22. Mairal J, Sapiro G, Elad M (2008) Learning multiscale sparse representations for image and video restoration. SIAM Multiscale Model Simul 7(1):214–241
23. Masnou S, Morel J.-M. (1998) Level lines based disocclusion. In: 5th IEEE international conference on image processing, Chicago, IL, Oct 4–7
24. Ogden JM, Adelson EH, Bergen JR, Burt PJ (1985) Pyramid-based computer graphics. RCA Eng 30(5):4–15
25. Patwardhan KA, Sapiro G, Bertalmío M (2007) Video inpainting under constrained camera motion. IEEE Trans Image Process 16(2):545–553
26. Shiratori T, Matsushita Y, Tang X, Kang SB (2006) Video completion by motion field transfer. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR), New York, vol 1
27. Sun J, Yuan L, Jia J, Shum HY (2005) Image completion with structure propagation. In: ACM SIGGRAPH 2005 papers. ACM, New York, p 868
28. Tschumperlé D (2006) Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. Int J Comput Vis 68(1):65–82
29. Wexler Y, Shechtman E, Irani M (2004) Space-time video completion. In: IEEE computer society conference on computer vision and pattern recognition (CVPR), Washington, DC, vol 1

Interactive Segmentation

Yuri Boykov
Department of Computer Science, University of Western Ontario, London, ON, Canada

Synonyms

Labeling; Object extraction; Partitioning; Segmentation; Semiautomatic; User-assisted; User-guided

Related Concepts

▶Dynamic Programming

Definition

Interactive image segmentation is a (near) real-time mechanism for accurately marking/labeling an object of interest based on visual user interface (VUI) specifying seeds, rough delineation, partial labeling, bounding box, or other constraints. Semiautomatic interactive segmentation methods incorporate various generic image cues and/or object-specific feature detectors in order to facilitate acceptable results with minimum user efforts.

Background

The most basic object extraction techniques like *thresholding* (Fig. 1) and *region growing* are based on simple but very fast heuristics. The spectrum of applications for such techniques is limited as they are prone to many problems, most notably *leaking* as in Fig. 2. Despite significant problems, thresholding and region growing are widely known due to their simplicity and speed. For example, they could be easily run on personal computers available 15–20 years ago. More recent generations of commodity PCs allow much more robust segmentation techniques, which rely on optimization of some segmentation cost function, or an energy. An energy functional should define an explicit measure of goodness for evaluating any specific segmentation result. The main goal of optimization is to find the best segmentation with respect to the specified criteria. In the context of interactive segmentation, the energy can embed some *soft* and *hard* constraints specified by the user.

Discrete Segmentation Functionals

Many discrete optimization methods for interactive segmentation are based on classical combinatorial optimization techniques: *dynamic programming* (DP) or *s/t graph cuts*. In general, these approaches are guaranteed to find the exact global minimum solution in finite (low-order polynomial) number of steps. There are no numerical convergence issues (e.g., oscillations), and they work in near real time even on a single CPU. For efficiency, these methods are often implemented using the simplest 4-neighbor grids. In theory, this basic approach may generate some discrete metrication artifacts, but they are rarely observable on real images. Increasing the neighborhood size (e.g., to eight neighbors) adequately addresses the problem [8, 9].

Graph-path segmentation models are designed for 2D image segmentation. *Intelligent scissors* [2], also known as *live wire* in the medical imaging community [3], requires user to place seeds on the desired object boundary; see Fig. 3. The algorithm connects these seeds by computing the shortest path on a graph where edges (or nodes) are image weighted according to local contrast (intensity gradient). Such weighting makes paths "stick" to image boundaries. The shortest paths from each new seed to all other image pixels can be pre-computed in $O(n \log n)$ time. Then, an optimal path from any mouse position to the seed can be previewed in real time.

This method evaluates segmentation boundary as a path between two seeds (see Fig. 4a) using energy functionals like

$$E(\mathcal{P}_{s,t}) = \sum_{\{p,q\}\in\mathcal{P}_{s,t}} w_{pq} \tag{1}$$

where $\mathcal{P}_{s,t}$ is a set of adjacent edges from source seed s to terminal seed t and edge weights $w_{pq} \geq 0$ are segmentation boundary costs based on some local measure of intensity contrast across edge $\{p,q\}$. One example of weights w_{pq} is

$$w_{pq} \propto \frac{1}{1 + |\nabla I \cdot n_{pq}|^2} \cdot ||p - q||$$

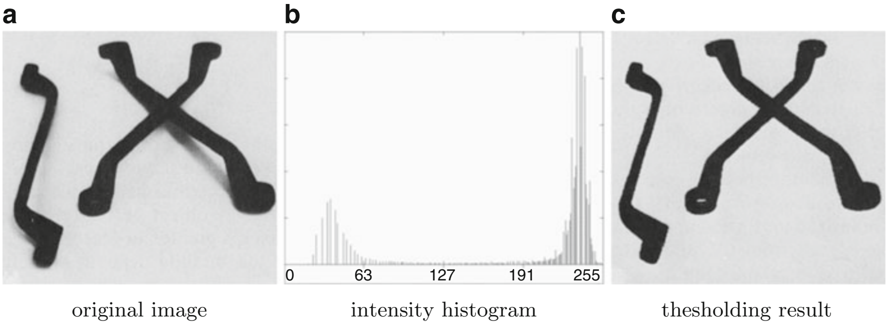
where ∇I is image gradient, vector n_{pq} is a normal to edge $\{p,q\}$, and $||p-q||$ is the geometric length of edge $\{p,q\}$. Factor $||p-q||$ differentiates diagonal edges from horizontal and vertical edges on grids with higher connectivity, which can reduce the grid bias.

There is a number of other interactive segmentation methods based on efficient DP-based optimization algorithms. For example, the methods in [10, 11] compute globally optimal cycles (closed contours), minimizing ratios of different measures of segment's boundary and region. For example, [11] can find a segment with the largest average contrast on its boundary. (Ratio of some cumulative contrast measure and the boundary length.) Optimization of ratio functionals evaluating boundary's curvature was addressed in [12].

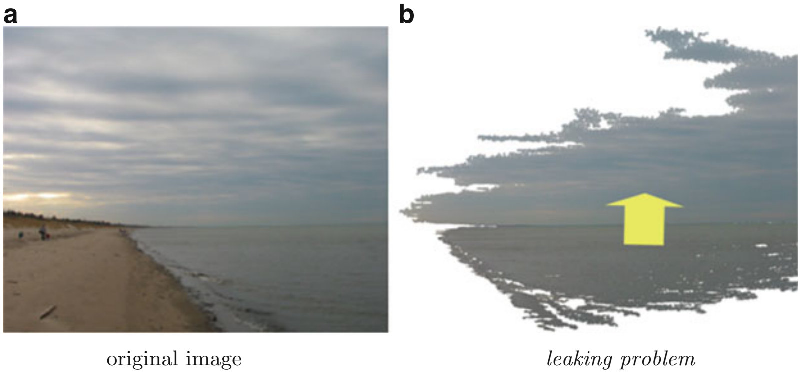
Graph-cut segmentation models: Boykov and Jolly [5] and Boykov and Funka-Lea [6] proposed an object extraction functional for N-dimensional images that evaluates boundary and region properties of segments as

$$E(x|\theta) = -\sum_{p:x_p=0} \ln \Pr(I_p|\theta_0) - \sum_{p:x_p=1} \ln \Pr(I_p|\theta_1) + \sum_{\{p,q\}\in\mathcal{N}} w_{pq} \cdot [x_p \neq x_q] \tag{2}$$

where $[\cdot]$ are *Iverson* brackets, variables x_p are binary object/background labels at pixels p, parameters



Interactive Segmentation, Fig. 1 Image *thresholding* segments a subset of pixels with intensities in a certain range, for example, $\{p : I_p < T\}$ in (**c**). Some threshold T works only if there is no overlap between the object and background intensities. The images above are from [1]



Interactive Segmentation, Fig. 2 *Region growing* is a greedy heuristic often associated with the *leaking* problem. Segment S is initialized by some *seed* in the object of interest (lake). Adjacent pixels q are iteratively added to S as long as some "growing" criteria are met, for example, $||I_q - I_p|| < T$ for some neighbor $p \in S$. A single low-contrast spot on the object boundary (horizon) will make the lake *leak* into the sky (**b**)

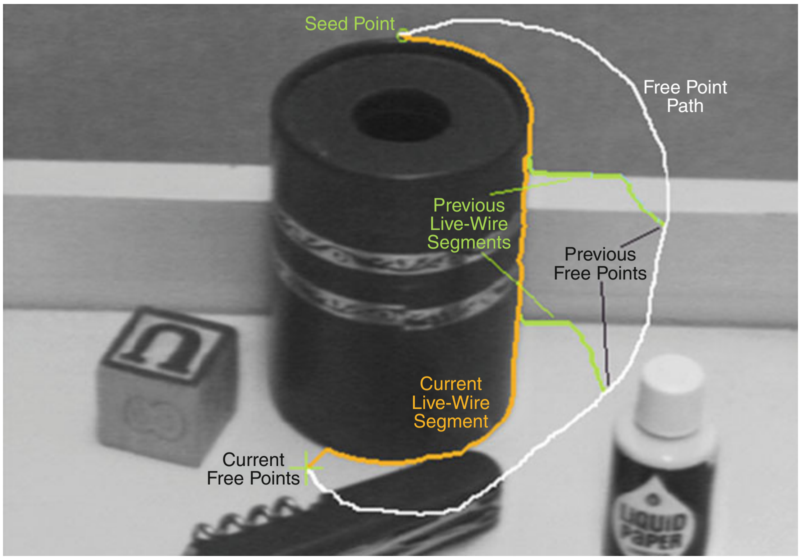
$\theta = \{\theta_0, \theta_1\}$ define object and background intensity distributions, and edge weights w_{pq} are a cost of discontinuity between a pair of neighboring pixels. For example,

$$w_{pq} \propto \exp\left(-\frac{|\nabla I \cdot e_{pq}|^2}{\sigma^2}\right) \cdot \frac{1}{||p-q||}$$

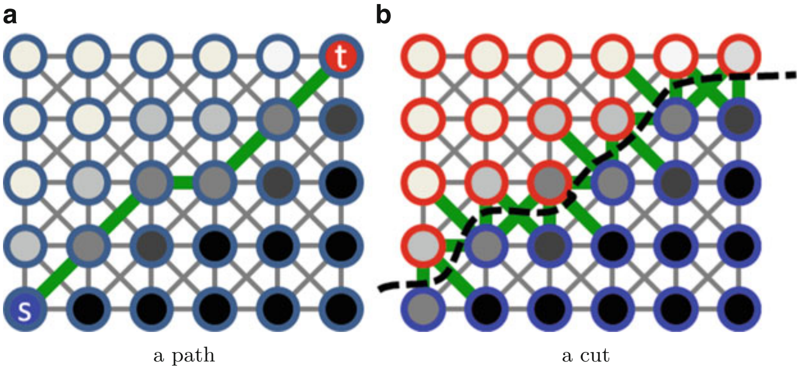
where e_{pq} is a unit vector collinear to edge $\{p, q\}$ and σ is a parameter controlling sensitivity to intensity contrast that is often set according to the level of noise in the image. Similarly to energy (1), the last term in (2) evaluates the image-weighted length of the segmentation boundary. In contrast to edge weights in (1), the weights above are based on intensity contrast along the edge $\{p, q\}$; see Fig. 4b. Normalization by edge length $||p-q||$ is required for grids with higher connectivity, reducing the grid bias [13].

The first two terms in (2) evaluate how well pixel intensities inside the object and background segments fit the corresponding distributions. In general, image intensity/color distributions could be extended by more sophisticated features and appearance models, for example, texture. The appearance models could be estimated from seeds or from prior data. The *grab-cut* method in [7] also uses an iterative EM-style scheme for additionally optimizing functional $E(x|\theta)$ in (2) with respect to parameters θ. In this case, sufficiently good initial appearance models can be often estimated from a user-placed box around the object. The graph-cut segmentation model also extends to video; for example, see the *snap-cut* method [14].

Functional (2) can be globally minimized over binary variables x by low-order polynomial algorithms from combinatorial optimization [15] that are fast even on a single CPU. Also, there are efficient



Interactive Segmentation, Fig. 3 *Intelligent scissors* or *live-wire* methods connect seeds placed on object boundaries. Optimal segmentation boundary (*orange curve*) can be previewed for any mouse position in real time



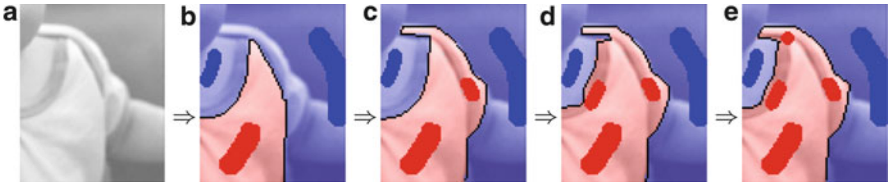
Interactive Segmentation, Fig. 4 Segmentation on graphs. The path-based methods [2–4] represent segmentation boundary as a sequence of adjacent (*green*) edges (**a**). A path could connect two seeds or form a closed cycle. The graph-cut methods [5–7] assign to pixels different labels, for example, *red* and *blue* in (**b**). Any such labeling implicitly defines a segmentation boundary, which is a *cut*, as a collection of (*green*) edges between differently labeled pixels

techniques [5, 6] for integrating interactive hard constraints (seeds) as in Fig. 5. Instead of segmentation energy (2), graph-cut framework can also use various ratio functionals [16].

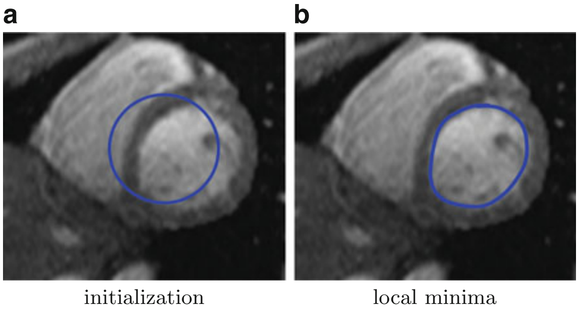
Segmentation energy (2) works for N objects (labels). In general, its optimization is NP-hard for $N > 2$. An approximate solution with a factor of 2 optimality guarantee can be found via α-expansion optimization algorithm [17]. Interestingly, imposing some additional geometric constraints between object boundaries (e.g., inclusion, exclusion, minimum margin) may lead to exact polynomial optimization algorithms [18–20].

Continuous Segmentation Functionals

Many popular interactive segmentation methods use continuous representation of segments where boundaries are contours in $\mathcal{R}^2$ or surfaces in $\mathcal{R}^3$. Such methods use either physics-based or geometric functionals to evaluate such continuous segments. Traditionally, variational calculus and different forms of *gradient descent* were used to converge to a local minima from a given initial contour; see Fig. 6. This motivates the general term, *active contours*, commonly used for such methods. Recent convex formulations for standard continuous regularization functionals [21, 22]



Interactive Segmentation, Fig. 5 Interactive editing of segments via hard constraints (seeds) based on graph cuts [5, 6]. A fragment of an original photo is shown in (**a**). Initial seeds and segmentation are shown in (**b**). The results in (**c**)–(**e**) illustrate changes in optimal segmentation as new hard constraints are successively added. The computation time for consecutive corrections in (**c**)–(**e**) is marginal compared to time for initial results in (**b**)



Interactive Segmentation, Fig. 6 *Snakes* and other *active contour* methods are initialized by rough delineation of the desired object (**a**). Minimization of an energy associated with the contour leads to a local minimum (**b**) with better alignment to image boundaries

and development of continuous max-flow techniques [23, 24] now also allow good quality approximations of the global minima.

Physics-based segmentation models: Snakes [25, 26], balloons [27], spline snakes [28], and other methods explicitly represent object boundaries as an elastic band or a balloon. The band is normally assigned an internal energy (elasticity and stiffness) and a potential energy with respect to some external field of predefined *image forces* attracting the band to image boundaries, that is, locations with large intensity gradients. A user can also place seeds defining additional attraction or repulsion potentials.

Geometry-based segmentation models: Note that two visually identical *snakes* appearing in the same image position may have different internal elastic energies. In many cases this may contradict a natural assumption that a segmentation result can be evaluated only by its visible appearance (In some video applications the goal is to track specific points on a moving segment, e.g. muscles of a beating heart. Physics-based (e.g. elastic) segmentation energy is well motivated is such cases). Based on this criticism of the physics-based approach, [29, 30] proposed *geodesic active contour* model evaluating contour C on a (bounded) domain Ω via geometric functionals like

$$E(C) = \int_{int(C)} f_1(p)\,dp + \int_{\Omega/int(C)} f_0(p)\,dp + \int_{\partial C} g(s)ds \tag{3}$$

which is similar to discrete model (2). The first two integrals in (3) are over the interior and exterior regions of C, and the third integral is the Riemannian length of C under metric g. Image-based *density* function g, for example,

$$g(p) = \frac{1}{1 + ||\nabla I(p)||^2}, \quad \forall p \in \Omega$$

shortens the length of contour C if it follows image boundaries where the density is small. The geometric length term in (3) is a continuous analogue of (1) and the spatial smoothness term in (2); see [13].

Scalar functions f_1 and f_0 on Ω are interior and exterior potentials based on some known appearance models for the object and background regions. For example, one can use $f_i(p) = -\ln \Pr(I_p|\theta_i)$ as in (2). Similarly, these potentials could also enforce user-placed hard constraints (seeds).

Geometric contours C can be represented as *level sets* of some scalar embedding function $u : \Omega \to \mathcal{R}^1$, for example, $C = \{p \in \Omega : u(p) = const\}$. This approach avoids some numerical issues, for example, the need for *reparameterization*, often associated with explicit representation of contour points needed for physics-based bands. The level-set framework does not pose any topological constraints on contours and yields easy-to-implement gradient descent equations for geometric energies like (3). More recently, geometric energies like (3) are addressed with various continuous convex formulations [24, 31, 32] that are shown to converge to a good approximation of the global minimum.

The continuous geometric models are very closely related to discrete segmentation energies in the graph-cut framework [8, 13]. One theoretical advantage of the continuous formulations is absence of the grid bias. Continuous numerical schemes guarantee certain convergence rate, but some stopping threshold often needs to be chosen. Current fast implementations of continuous optimization methods, for example, [32], require GPU acceleration.

Distance-Based Segmentation Methods

Many interactive segmentation techniques optimize objective functions that are only indirectly related to the visual appearance of the segments and their boundaries. In particular, a large number of methods compute optimal (image-weighted) distance functions computed from seeds. For example, *fast-marching method* [33] extracts the boundary reached at time T by a front expanding with an image-weighted speed. This can be seen as a generalization of *region growing*. These ideas were extended in [34] where the segments are Voronoi cells w.r.t. geodesic distance $d(p,s)$ from the object and background seeds $s \in S_O \cup S_B$

$$x_p^* = \arg \min_{l \in \{O,B\}} \min_{s \in S_l} d(p,s)$$

Their image-based metric is based on gradients of the appearance models likelihoods instead of intensity gradients. Distance transforms can also work as the unary potentials in the segmentation models (2) and (3); for example, see [35].

Instead of the standard *min-sum* geodesic distance $d(p,s)$, many segmentation methods use other measures to compute Voronoi cells from the seeds. For example, *fuzzy-connectedness* methods [36, 37] compute Voronoi cells with respect to some *max–min* affinity measure. *Random walker* [38] outputs Voronoi cells for probabilistic distance function $d(p,s)$, measuring the expected time of arrival for a random walk from p to s. *Watershed* method, for example, [39], connect points to seeds using *water-drop paths* instead of geodesics. *Power watershed* algorithm [40] unifies the ideas of *watershed* and *random walker*.

Some Open Problems

Energy functionals like (2) and (3) represent only the most standard ideas for evaluating segments. Accurate evaluation of the higher-order geometric properties of the boundary, for example, curvature [12, 41], remains a difficult optimization issue. Shape priors for globally optimal segmentation [42, 43] as well as enforcement of topological constraints [44, 45] are largely open problems.

References

1. Gonzalez RC, Woods RE (2007) Digital image processing, 3rd edn. Prentice Hall, Harlow
2. Mortensen EN, Barrett WA (1998) Interactive segmentation with intelligent scissors. Graph. Models Image Process 60:349–384
3. Falcão AX, Udupa JK, Samarasekera S, Sharma S (1998) User-steered image segmentation paradigms: live wire and live lane. Graph Models Image Process 60:233–260
4. Jermyn IH, Ishikawa H (1999) Globally optimal regions and boundaries. In: International conference on computer vision, Kerkyra, vol II, pp 904–910
5. Boykov Y, Jolly MP (2001) Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: International conference on computer vision, Vancouver, vol II, pp 105–112
6. Boykov Y, Funka-Lea G (2006) Graph cuts and efficient N-D image segmentation. Int J Comput Vis 70(2):109–131
7. Rother C, Kolmogorov V, Blake A (2004) Grabcut – interactive foreground extraction using iterated graph cuts. ACM Trans Graph 23:307–331
8. Kolmogorov V, Boykov Y (2005) What metrics can be approximated by geo-cuts, or global optimization of length/area and flux. In: International conference on computer vision, Beijing
9. Boykov Y, Kolmogorov V, Cremers D, Delong A (2006) An integral solution to surface evolution PDEs via geo-cuts. In: European conference on computer vision (ECCV), Graz, Austria
10. Cox IJ, Rao SB, Zhong Y (1996) Ratio regions: a technique for image segmentation. In: International conference on pattern recognition, Vienna, vol II, pp 557–564

11. Jermyn IH, Ishikawa H (2001) Globally optimal regions and boundaries as minimum ratio weight cycles. PAMI 23(10):1075–1088
12. Schoenemann T, Cremers D (2007) Introducing curvature into globally optimal image segmentation: minimum ratio cycles on product graphs. In: International conference on computer vision (ICCV), Rio de Janeiro
13. Boykov Y, Kolmogorov V (2003) Computing geodesics and minimal surfaces via graph cuts. In: International conference on computer vision, Nice, vol I, pp 26–33
14. Bai X, Wang J, Simons D, Sapiro G (2009) Video SnapCut: Robust video object cutout using localized classifiers. In: ACM transactions on graphics (SIGGRAPH), Yokohama
15. Boykov Y, Kolmogorov V (2004) An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans Pattern Anal Mach Intell 26(9):1124–1137
16. Kolmogorov V, Boykov Y, Rother C (2007) Applications of parametric maxflow in computer vision. In: International conference on computer vision (ICCV), Rio de Janeiro
17. Boykov Y, Veksler O, Zabih R (2001) Fast approximate energy minimization via graph cuts. IEEE Trans Pattern Anal Mach Intell 23(11):1222–1239
18. Li K, Wu X, Chen DZ, Sonka M (2006) Optimal surface segmentation in volumetric images-a graph-theoretic approach. IEEE Trans Pattern Anal Pattern Recognit (PAMI) 28(1):119–134
19. Delong A, Boykov Y (2009) Globally optimal segmentation of multi-region objects. In: International conference on computer vision (ICCV), Kyoto
20. Felzenszwalb PF, Veksler O (2010) Tiered scene labeling with dynamic programming. In: IEEE conference on computer vision and pattern recognition (CVPR), San Francisco
21. Chan T, Esedoglu S, Nikolova M (2006) Algorithms for finding global minimizers of image segmentation and denoising models. SIAM J Appl Math 66(5):1632–1648
22. Pock T, Chambolle A, Cremers D, Bischof H (2009) A convex relaxation approach for computing minimal partitions. In: IEEE conference on computer vision and pattern recognition (CVPR), Miami
23. Appleton B, Talbot H (2006) Globally minimal surfaces by continuous maximal flows. IEEE Trans Pattern Anal Pattern Recognit 28(1):106–118
24. Yuan J, Bae E, Tai XC (2010) A study on continuous max-flow and min-cut approaches. In: IEEE conference on computer vision and pattern recognition (CVPR), San Francisco
25. Kass M, Witkin A, Terzolpoulos D (1988) Snakes: active contour models. Int J Comput Vis 1(4):321–331
26. Amini AA, Weymouth TE, Jain RC (1990) Using dynamic programming for solving variational problems in vision. IEEE Trans Pattern Anal Mach Intell 12(9):855–867
27. Cohen LD, Cohen I (1993) Finite element methods for active contour models and balloons for 2-d and 3-d images. IEEE Trans Pattern Anal Mach Intell 15(11):1131–1147
28. Isard M, Blake A (1998) Active contours. Springer, London
29. Caselles V, Kimmel R, Sapiro G (1997) Geodesic active contours. Int J Comput Vis 22(1):61–79
30. Yezzi A Jr, Kichenassamy S, Kumar A, Olver P, Tannenbaum A (1997) A geometric snake model for segmentation of medical imagery. IEEE Trans Med Imaging 16(2):199–209
31. Unger M, Pock T, Cremers D, Bischof H (2008) Tvseg - interactive total variation based image segmentation. In: British machine vision conference (BMVC), Leeds, UK
32. Santner J, Pock T, Bischof H (2010) Interactive multi-label segmentation. In: Asian conference on computer vision (ACCV), Queenstown
33. Malladi R, Sethian J (1998) A real-time algorithm for medical shape recovery. In: International conference on computer vision (ICCV), Bombay, pp 304–310
34. Bai X, Sapiro G (2007) A geodesic framework for fast interactive image and video segmentation and matting. In: IEEE international conference on computer vision (ICCV), Rio de Janeiro
35. Criminisi A, Sharp T, Blake A (2008) Geos: geodesic image segmentation. In: European conference on computer vision (ECCV), Marseille
36. Udupa JK, Samarasekera S (1996) Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. Graph Models Image Process 58(3):246–261
37. Herman GT, Carvalho BM (2001) Multiseeded segmentation using fuzzy connectedness. IEEE Trans Pattern Anal Mach Intell (PAMI) 23(5):460–474
38. Grady L (2006) Random walks for image segmentation. IEEE Trans Pattern Anal Pattern Recognit (PAMI) 28(11):1768–1783
39. Cousty J, Bertrand G, Najman L, Couprie M (2009) Watershed cuts: minimum spanning forests and the drop of water principle. IEEE Trans Pattern Anal Mach Intell (PAMI) 31(8):1362–1374
40. Couprie C, Grady L, Najman L, Talbot H (2011). Power watersheds: a unifying graph based optimization framework. IEEE transactions on Pattern Analysis and Machine Intelligence (PAMI) 33(7):1384–1399
41. Williams DJ, Shah M (1992) A fast algorithm for active contours and curvature estimation. Comput Vis Graph Image Process 55(1):14–26
42. Veksler O (2008) Star shape prior for graph-cut image segmentation. In: European conference on computer vision (ECCV), Marseille
43. Felzenszwalb P, Veksler O (2010) Tiered scene labelling with dynamic programming. In: IEEE conference on computer vision and pattern recognition (CVPR), San Francisco
44. Vicente S, Kolmogorov V, Rother C (2008) Graph cut based image segmentation with connectivity priors. In: IEEE conference on computer vision and pattern recognition (CVPR), Anchorage
45. Nowozin S, Lampert CH (2009) Global connectivity potentials for random field models. In: IEEE conference on computer vision and pattern recognition (CVPR), Miami

Interface Reflection

►Specularity, Specular Reflectance

Interpretation of Line Drawings

▶Line Drawing Labeling

Interreflections

Michael S. Langer
School of Computer Science, McGill University,
Montreal, QC, Canada

Synonyms

Mutual illumination

Related Concepts

▶Bas-Relief Ambiguity; ▶Diffuse Reflectance; ▶Radiance

Definition

Interreflections are reflections of light from one surface to another surface.

Background

Surfaces are illuminated not just by light sources but also by each other. These interreflections can provide a significant component of surface illumination, especially in concavities or enclosures. Numerical methods for computing interreflections were developed in the early twentieth century to solve problems in heat transfer such as in furnace design. The methods were developed further by the computer graphics community in the 1980s to render global illumination for scenes with Lambertian surfaces and later for scenes with specular components [7, 8].

Theory

Interreflections can be described mathematically in several equivalent ways. One way is to write the reflected light as a sum of the light that is due to the illumination that arrives at a surface directly from the light source, plus the light that arrives from other surfaces in the scene via interreflections. Suppose the scene is composed of Lambertian surfaces with albedo $\rho(\mathbf{x})$ varying across surfaces. Let $L_s(\mathbf{x})$ be the component of $\mathbf{x}$'s outgoing radiance that is due to direct illumination from the source. Then the total radiance $L(\mathbf{x})$ leaving $\mathbf{x}$ is the sum of $L_s(\mathbf{x})$ and the radiance that is due to interreflections:

$$L(\mathbf{x}) = L_s(\mathbf{x}) + \rho(\mathbf{x}) \int_{\mathcal{S}} L(\mathbf{y}) K(\mathbf{x}, \mathbf{y}) d\mathbf{y}. \quad (1)$$

Here the integral is taken over all surface points $\mathbf{y} \in \mathcal{S}$ in the scene, and the function $K(\mathbf{x}, \mathbf{y})$ is a symmetric weighting function that depends on the surface normals at $\mathbf{x}$ and $\mathbf{y}$ and on the distances between $\mathbf{x}$ and $\mathbf{y}$. $K(\mathbf{x}, \mathbf{y})$ is zero if the $\mathbf{x}$ and $\mathbf{y}$ are not visible to each other.

It is common to approximate Eq. (1) by using a polygonal mesh surface whose facets have constant radiance:

$$\mathbf{r} = \mathbf{r}_s + \mathbf{PKr} \quad (2)$$

where $\mathbf{r}$ and $\mathbf{r}_s$ are the vectors of total and direct radiance, respectively, $\mathbf{P}$ is a diagonal matrix of albedos, and $\mathbf{K}$ is called the *form factor* matrix. The above equations can be generalized to non-Lambertian surfaces as well [7, 8].

A second approach is to consider the eigenfunctions of $\mathbf{K}$ which are radiance functions that are invariant to interreflections [9, 11]. These eigenfunctions are concentrated in surface concavities and at points of contact between surfaces [10].

A third approach is to use ray tracing to follow the light emitted from the source through successive reflections or bounces in the scene. The nth reflection serves as the source for the $n+1$st reflection, and the sum of all reflections gives an infinite series. For any scene geometry and reflectance, it is possible to construct a linear operator that can be applied iteratively to decompose the interreflections into their n bounce components [15]. Understanding the various bounces is especially important for making finite approximations. For example, a two-bounce model has been used to model how surface microfacets can account for non-Lambertian reflection [14] and how color bleeding occurs between surfaces in a concavity [5].

Application

Standard shape from shading and photometric stereo methods consider only the direct illumination component [6]. When interreflections are present, these methods produce erroneous results [4]. It is possible to extend these methods to account for interreflections by first ignoring interreflections to obtain an approximate solution and then iteratively updating the solution to account for interreflections. This idea has been applied to photometric stereo [12] and to shape from shading for the special case that the surface is an unfolded book in a photocopier [16].

The above applications assume the light sources are known. But what if they are unknown? An important fact that applies in this situation is the *bas-relief ambiguity* [1]. When a Lambertian surface is illuminated by direct illumination only, there exists a family of scenes (shape, albedo, lighting) that all produce the same image. With interreflections present, the bas-relief ambiguity no longer exists [3]. One can estimate the surface shape and reflectance similarly to above, namely, by applying a photometric stereo method that is designed for unknown lighting [17] and then iteratively updating the shape and reflectance to account for interreflections [3].

Interreflections also arise in projector-camera systems. An image that is projected on a concave screen will suffer from interreflections that will lower the image contrast. This contrast reduction can be compensated for, to some extent, by modifying the originally projected pattern [2]. A related example which involves active illumination of a 3D scene is to obtain a small number of images of the scene by illuminating it with a set of high-frequency projection patterns such as checkerboards [13]. The interreflection components of the scene will have relatively low spatial frequency and will be similar in the images. This property allows one to decompose the image of a fully illuminated scene into its direct component and its interreflection component. Unlike most methods for interreflections which assume Lambertian scenes, this method allows for non-Lambertian surfaces and other forms of reflections such as volume scattering.

References

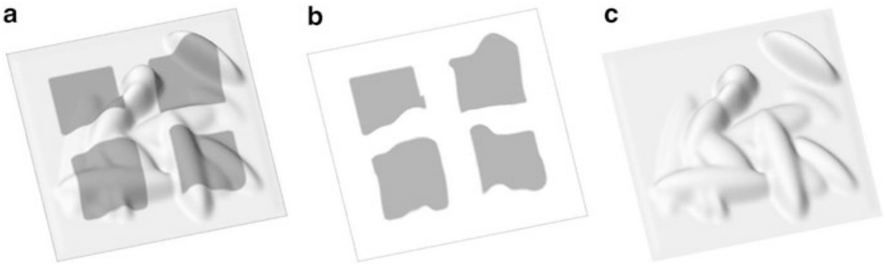
1. Belhumeur PN, Kriegman DJ, Yuille AL (1999) The bas-relief ambiguity. Int J Comput Vis 35(1):33–44
2. Bimber O, Grundhofer A, Zeidler T, Danch D, Kapakos P (2006) Compensating indirect scattering for immersive and semi-immersive projection displays. In: Virtual reality conference 2006, Alexandria, pp 151–158, 25–29
3. Chandraker MK, Kahl F, Kriegman DJ (2005) Reflections on the generalized bas-relief ambiguity. In: CVPR ’05: proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR), Washington, DC, USA. IEEE Computer Society, pp 788–795
4. Forsyth D, Zisserman A (1991) Reflections on shading. IEEE Trans Pattern Anal Mach Intell 13:671–679
5. Funt BV, Drew MS, Ho J (1991) Color constancy from mutual reflection. Int J Comput Vis 6(1):5–24
6. Horn BKP (1977) Understanding image intensities. Artif Intell 8(2):201–231
7. Immel DS, Cohen MF, Greenberg DP (1986) A radiosity method for non-diffuse environments. ACM Trans Graph 20(4):133–142
8. Kajiya JT (1986) The rendering equation. ACM Trans Graph 20(4):143–150
9. Kœnderink JJ, van Doorn AJ (1983) Geometrical modes as a general method to treat diffuse interreflections in radiometry. J Opt Soc Am 73(6):843–850
10. Langer MS (1999) When shadows become interreflections. Int J Comput Vis 34(2/3):1–12
11. Moon P (1940) On interreflections. J Opt Soc Am 30: 195–205
12. Nayar SK, Ikeuchi K, Kanade T (1991) Shape from interreflections. Int J Comput Vis 6:173–195
13. Nayar SK, Krishnan A, Grossberg MD, Raskar R (2006) Fast separation of direct and global components of a scene using high frequency illumination. ACM Trans Graph 25:935–944
14. Nayar SK, Oren M (1995) Generalization of the Lambertian model and implications for machine vision. Int J Comput Vis 14(3):227–251
15. Seitz SM, Matsushita Y, Kutulakos KN (2005) A theory of inverse light transport. In: ICCV ’05: proceedings of the tenth IEEE international conference on computer vision, Washington, DC, USA. IEEE Computer Society, pp 1440–1447
16. Wada T, Ukida H, Matsuyama T (1997) Shape from shading with interreflections under a proximal light-source: distortion-free copying of an unfolded book. Int J Comput Vis 24(2):125–135
17. Yuille AL, Snow D, Epstein R, Belhumeur PN (1999) Determining generative models of objects under varying illumination: shape and albedo from multiple images using svd and integrability. Int J Comput Vis 35(3):203–222

Intrinsic Images

Marshall F. Tappen
University of Central Florida, Orlando, FL, USA

Related Concepts

▶Image Decompositions



Intrinsic Images, Fig. 1 These images show an example of an intrinsic image decomposition. In this decomposition, the intrinsic images can be how the image in (**a**) can be decomposed into the albedo and shading images shown in (**b**) and (**c**), respectively

Definition

A set of images used to represent characteristics of a scene pictured in an image, with each image representing one particular characteristic of the scene.

Background

Vision systems have been categorized into low- and high-level processing, with high-level processing taking an object-centered approach [1]. In this categorization, the role of low-level processing is to extract basic characteristics at all locations in the image. These characteristics are then used to find objects.

Intrinsic images are a method for representing the low-level characteristics extracted from images. In the intrinsic image representation, proposed by Barrow and Tenenbaum in [2], one image represents each of the characteristics being used in the system. The value of each pixel represents the value of the characteristic at each point in the scene.

The types of characteristics that are conveniently expressed as intrinsic images include the illumination of each point in the scene, the motion at each point, the orientation of each point, the albedo, and the distance from the camera.

Application

Starting with [3], the term intrinsic images have also been used to refer to an image decomposition that decomposes an observed image into intrinsic images that can be recombined to recreate the observed image. The most common decomposition is, into images, representing the shading, or illumination, and albedo of each point. Figure 1 shows an example of how the image in Fig. 1a into shading and albedo images. Mathematically, the decomposition is modeled as

$$O_p = I_p \times R_p$$

where O_p is the the value of the observed image at pixel p, I is the illumination image, and R is the reflectance image.

In [3], Weiss recovers these intrinsic images from a sequence of images where the illumination varies in the scene. In [4] and [5], Tappen et al. use color and gray-scale features to estimate the decomposition from a single image.

Besides image decompositions, [6] proposes using intrinsic images that represent properties like occlusion boundaries and object depth.

References

1. Szeliski R (1990) Bayesian modeling of uncertainty in low-level vision. Int J Comput Vis 5(3):271–301
2. Barrow HG, Tenenbaum JM (1978) Recovering intrinsic scene characteristics from images. In: Hanson A, Riseman E (eds) Computer vision systems. Academic, New York, pp 3–26
3. Weiss Y (2001) Deriving intrinsic images from image sequences. In: The proceedings of the IEEE international conference on computer vision, Vancouver, pp 68–75
4. Tappen MF, Freeman WT, Adelson EH (2005) Recovering intrinsic images from a single image. IEEE Trans Pattern Anal Mach Intell 27(9):1459–1472
5. Tappen MF, Adelson EH, Freeman WT (2006) Estimating intrinsic component images using non-linear regression. In: The Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR), vol 2. IEEE Computer Society, Los Alamitos, pp 1992–1999
6. Hoiem D, Efros AA, Hebert M (2008) Closing the loop on scene interpretation. in: Proceedings the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Piscataway

Intrinsic Parameters

▶Intrinsics

Intrinsics

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Synonyms

Intrinsic parameters

Related Concepts

▶Camera Parameters (Intrinsic, Extrinsic)

Definition

Intrinsics, short for *intrinsic parameters*, refer to the parameters belonging to the essential nature of a thing, which is usually a camera in computer vision. The intrinsic parameters of a camera include its focal length, the aspect ratio of a pixel, the coordinates of its principal point, and the lens distortion parameters.

See entry "▶Camera Parameters" for more details.

Inverse Compositional Algorithm

Simon Baker
Microsoft Research, Redmond, WA, USA

Definition

The inverse compositional algorithm is a reformulation of the classic Lucas-Kanade algorithm to make the steepest-descent images and Hessian constant.

Background: Lucas-Kanade

The goal of the Lucas-Kanade algorithm is to minimize the sum of squared error between a template image $T(\mathbf{x})$ and a warped input image $I(\mathbf{x})$:

$$\sum_{\mathbf{x}} [\, T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))\,]^2 , \qquad (1)$$

where $\mathbf{x} = (x, y)^{\mathrm{T}}$ are the pixel coordinates, $\mathbf{W}(\mathbf{x};\mathbf{p})$ is a parameterized set of warps, and $\mathbf{p} = (p_1, \ldots p_n)^{\mathrm{T}}$ is a vector of parameters. The Lucas-Kanade algorithm assumes that a current estimate of $\mathbf{p}$ is known and then iteratively solves for increments to the parameters $\Delta\mathbf{p}$, i.e., approximately minimize

$$\sum_{\mathbf{x}} [\, T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p} + \Delta\mathbf{p}))\,]^2 , \qquad (2)$$

with respect to $\Delta\mathbf{p}$ and update the parameters

$$\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}. \qquad (3)$$

Equation (2) is linearized by performing a first-order Taylor expansion:

$$\sum_{\mathbf{x}} \left[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) - \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \right]^2 . \qquad (4)$$

In this expression, $\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right)$ is the *gradient* of image I and $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is the *Jacobian* of the warp. Equation (4) has a closed-form solution as follows. The partial derivative of the expression in Eq. (4) with respect to $\Delta\mathbf{p}$ is

$$-2 \sum_{\mathbf{x}} \left[\nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \right]^{\mathrm{T}} \left[T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) - \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} \right] . \qquad (5)$$

Then denote

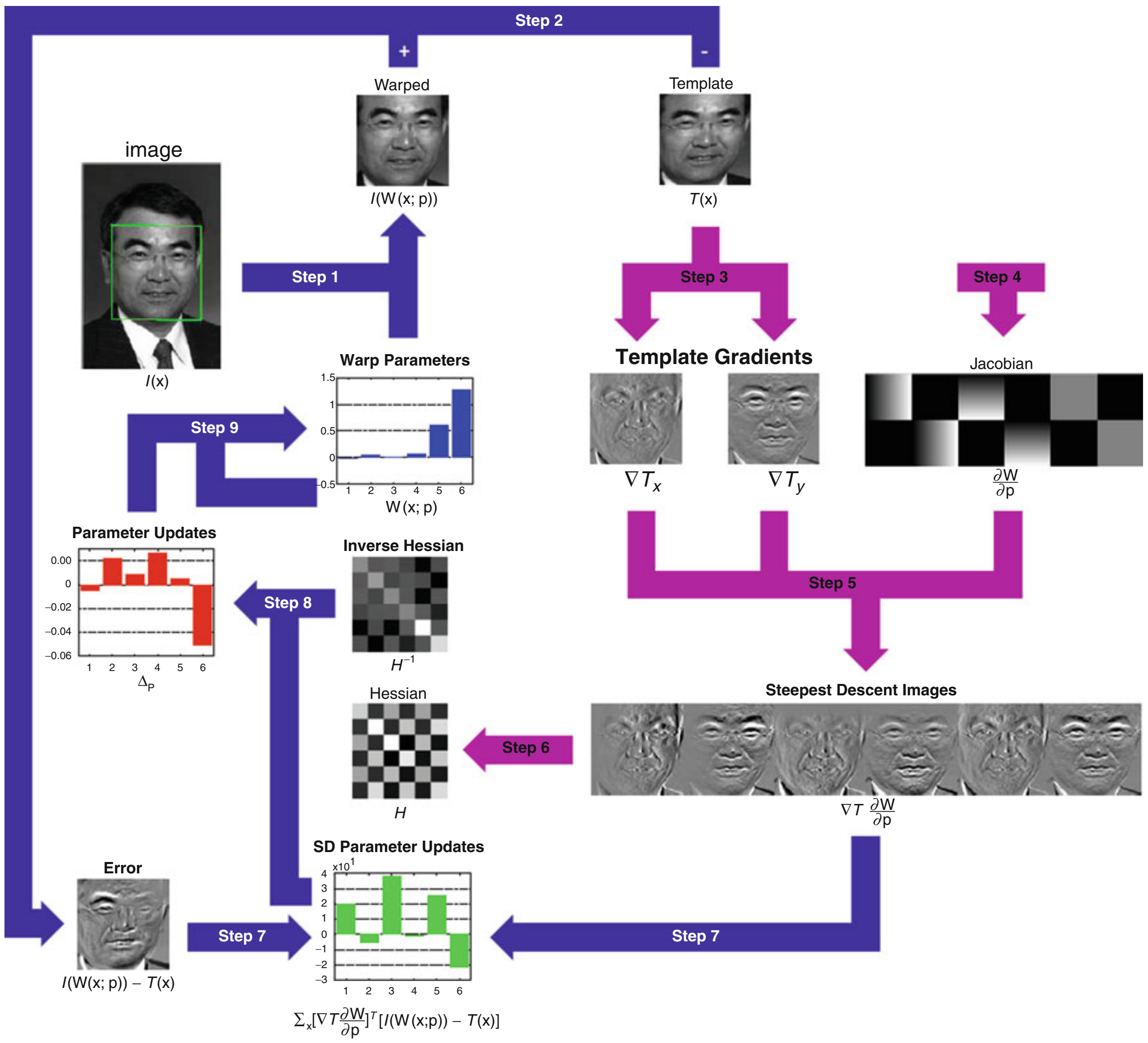
$$\mathbf{SD}_{\mathrm{lk}}(\mathbf{x}) = \nabla I \frac{\partial \mathbf{W}}{\partial \mathbf{p}} , \qquad (6)$$

the *steepest-descent* images. Setting the expression in Eq. (5) to equal zero and solving give

$$\Delta\mathbf{p} = H_{\mathrm{lk}}^{-1} \sum_{\mathbf{x}} \mathbf{SD}_{\mathrm{lk}}^{\mathrm{T}}(\mathbf{x})\, E(\mathbf{x}) \qquad (7)$$

where H_{lk} is the $n \times n$ (Gauss-Newton approximation to the) *Hessian* matrix

$$H_{\mathrm{lk}} = \sum_{\mathbf{x}} \mathbf{SD}_{\mathrm{lk}}^{\mathrm{T}}(\mathbf{x})\, \mathbf{SD}_{\mathrm{lk}}(\mathbf{x}) \qquad (8)$$



Inverse Compositional Algorithm, Fig. 1 A schematic overview of the inverse compositional algorithm. Steps 3–6 (*light-color arrows*) are performed once as a precomputation. The main algorithm simply consists of iterating image warping (Step 1), image differencing (Step 2), image dot products (Step 7), multiplication with the inverse of the Hessian (Step 8), and the update to the warp (Step 9). All of these steps can be performed efficiently

and

$$E(\mathbf{x}) = T(\mathbf{x}) - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \qquad (9)$$

is the *error image*. The Lucas-Kanade algorithm consists of iteratively applying Eqs. (7) and (3). Because the gradient ∇I must be evaluated at $\mathbf{W}(\mathbf{x};\mathbf{p})$ and the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ at $\mathbf{p}$, they both depend on $\mathbf{p}$. Both the steepest-descent images and the Hessian must therefore be recomputed in every iteration [1, 2].

The Inverse Compositional Algorithm

Baker and Matthews [3] proposed the inverse compositional algorithm as a way of reformulating image alignment so that the steepest descent images and Hessian are constant. Although the goal of the inverse compositional algorithm is the same as the Lucas-Kanade algorithm (e.g., minimizing Eq. (1)), the inverse compositional algorithm iteratively minimizes

$$\sum_{\mathbf{x}} [T(\mathbf{W}(\mathbf{x};\Delta\mathbf{p})) - I(\mathbf{W}(\mathbf{x};\mathbf{p}))]^2, \qquad (10)$$

with respect to $\Delta\mathbf{p}$ and then updates the warp

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \leftarrow \mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}. \tag{11}$$

The expression

$$\mathbf{W}(\mathbf{x};\mathbf{p}) \circ \mathbf{W}(\mathbf{x};\Delta\mathbf{p}) \equiv \mathbf{W}\left(\mathbf{W}(\mathbf{x};\Delta\mathbf{p});\mathbf{p}\right) \tag{12}$$

is the composition of 2 warps, and $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})^{-1}$ is the inverse of $\mathbf{W}(\mathbf{x};\Delta\mathbf{p})$. The inverse compositional algorithm iterates Eq. (10) and (11) and can be shown to be equivalent to the Lucas-Kanade algorithm to first order in $\Delta\mathbf{p}$ [3].

Performing a first-order Taylor expansion on Eq. (10) gives

$$\sum_{\mathbf{x}} \left[T(\mathbf{W}(\mathbf{x};\mathbf{0})) + \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}} \Delta\mathbf{p} - I(\mathbf{W}(\mathbf{x};\mathbf{p})) \right]^2. \tag{13}$$

Assuming that $\mathbf{W}(\mathbf{x};\mathbf{0})$ is the identity warp, the minimum of this expression is

$$\Delta\mathbf{p} = -H_{\text{ic}}^{-1} \sum_{\mathbf{x}} \mathbf{SD}_{\text{ic}}^{\text{T}}(\mathbf{x})\, E(\mathbf{x}), \tag{14}$$

where $\mathbf{SD}_{\text{ic}}^{\text{T}}(\mathbf{x})$ are the steepest-descent images with I replaced by T:

$$\mathbf{SD}_{\text{ic}}(\mathbf{x}) = \nabla T \frac{\partial \mathbf{W}}{\partial \mathbf{p}}, \tag{15}$$

H_{ic} is the Hessian matrix computed using the new steepest-descent images:

$$H_{\text{ic}} = \sum_{\mathbf{x}} \mathbf{SD}_{\text{ic}}^{\text{T}}(\mathbf{x})\, \mathbf{SD}_{\text{ic}}(\mathbf{x}), \tag{16}$$

and the Jacobian $\frac{\partial \mathbf{W}}{\partial \mathbf{p}}$ is evaluated at $(\mathbf{x};\mathbf{0})$. Since there is nothing in either the steepest-descent images or the Hessian that depends on $\mathbf{p}$, they can both be precomputed. The inverse composition algorithm is illustrated in Fig. 1.

Application

The inverse compositional algorithm can be used almost anywhere the Lucas-Kanade can be. In can be applied to anything from simple translational motion to dense optical flow. Perhaps the most significant application is its use to speed-up the fitting or tracking of active appearance models [4, 5].

References

1. Hager G, Belhumeur P (1998) Efficient region tracking with parametric models of geometry and illumination. IEEE Trans Pattern Anal Mach Intell 20(10):1025–1039
2. Shum HY, Szeliski R (2000) Construction of panoramic image mosaics with global and local alignment. Int J Comput Vis 16(1):63–84
3. Baker S, Matthews I (2004) Lucas-Kanade 20 years on: a unifying framework. Int J Comput Vision 56(3):221–255
4. Cootes T, Edwards G, Taylor C (2001) Active appearance models. IEEE Trans Pattern Anal Mach Intell 23(6):681–685
5. Matthews I, Baker S (2004) Active appearance models revisited. Int J Comput Vis 60(2):135–164

IP Camera

▸Pan-Tilt-Zoom (PTZ) Camera

Irradiance

Fabian Langguth and Michael Goesele
GCC - Graphics, Capture and Massively Parallel
Computing, TU Darmstadt, Darmstadt, Germany

Related Concepts

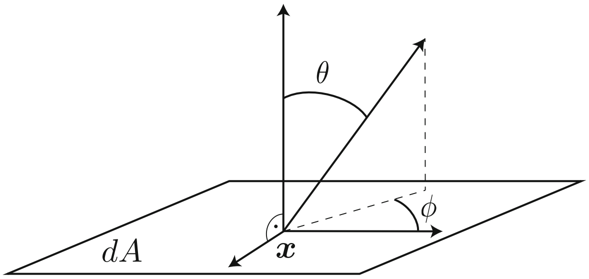
▸Radiance

Definition

Irradiance E is defined as the incident power of electromagnetic radiation on a surface per unit surface area. It is expressed in watt per square meter ($\text{W} \cdot \text{m}^{-2}$).

Background

Irradiance is a concept from radiometry, the science of measuring radiant energy transfer [1]. The equivalent



Irradiance, Fig. 1 Geometric setting

concept in photometry is illuminance, with the key difference being that illuminance is adjusted to account for the varying sensitivity of the human eye to different wavelengths of light.

Theory

The irradiance at a surface point $\boldsymbol{x}$ is proportional to the radiance $L(\boldsymbol{x}, \theta, \phi)$ arriving at $\boldsymbol{x}$ from direction (θ, ϕ) with a geometric foreshortening factor $\cos\theta$. Taking into account the whole hemisphere above the surface point, the irradiance is the integral over all incoming directions

$$E(\boldsymbol{x}) = \int_{\theta,\phi} L(\boldsymbol{x}, \theta, \phi) \cos\theta \;\; d\theta d\phi. \qquad (1)$$

θ denotes the angle between the surface normal and the incident direction (θ, ϕ) (see Fig. 1).

Application

For a camera with an optical lens and an aperture, the image irradiance at a camera sensor is proportional to the radiance L emitted from a small scene patch in the form that

$$E = L\frac{\pi}{4}\left(\frac{d}{f}\right)^2 \cos\alpha^4 \qquad (2)$$

where d is the aperture and f the focal length of the lens. α is the angle between the direction to the observed patch and the principal ray of the camera. For wide-angle lenses, the influence of α often results in a reduction of an image's brightness at the corners compared to the image center. This effect is also called vignetting.

The pixel values of digital images are directly related to the irradiance at the sensor of the camera via the camera's response curve [2, 3]. Many computer vision techniques such as photometric stereo use this fact to recover information about the scene from the irradiance. Early works in this field include [4] and [5].

References

1. Dutré P, Bala K, Bekaert P (2006) Advanced global illumination. AK Peters, Wellesley
2. Debevec PE, Malik J (1997) Recovering high dynamic range radiance maps from photographs. In: Proceedings of the 24th annual conference on computer graphics and interactive techniques, SIGGRAPH '97, Los Angeles. ACM/Addison-Wesley, Los Angeles, pp 369–378
3. Robertson MA, Borman S, Stevenson RL (2003) Estimation-theoretic approach to dynamic range enhancement using multiple exposures. J Electron Imaging 12(2):219
4. Bruss AR (1981) The image irradiance equation: its solution and application (AITR-623). Artificial Intelligence Laboratory, Massachusetts Institute of Technology
5. Woodham R (1980) Photometric method for determining surface orientation from multiple images. Opt Eng 19(1):139–144

Isotropic Differential Geometry in Graph Spaces

Jan J. Koenderink
Faculty of EEMSC, Delft University of Technology,
Delft, The Netherlands
The Flemish Academic Centre for Science and the
Arts (VLAC), Brussels, Belgium
Laboratory of Experimental Psychology, University
of Leuven (K.U. Leuven), Leuven, Belgium

Synonyms

Differential Geometry of Graph Spaces; Dual Differential Geometry

Related Concepts

▸Curvature; ▸Curves in Euclidean Three-Space; ▸Isotropic Differential Geometry in Graph Spaces

Definition

In many settings the conventional Euclidean differential geometry is not appropriate. A common case involves "graphs," an instance being images where the carrier may be modeled as the Euclidean plane, but the intensity domain is incommensurate. Isotropic differential geometry allows one to deal with such cases.

Background

Isotropic differential geometry became highly developed during the first half of the twentieth century, mainly in German-speaking countries. The bulk of the literature is still in German.

Theory

There are frequent cases in computer vision and image processing in which the Euclidean $\mathbb{E}^3$ setting from classical differential geometry is not appropriate. A simple example is an image, which may be thought of as the Euclidean plane $\mathbb{E}^2$, augmented with some "intensity domain." Intensities are nonnegative quantities that somehow reflect photon-catches in, e.g., CCD devices. Usually the physical dimension is unclear and considered irrelevant to the problem. Then the structure of the intensity domain is most appropriately modeled by the affine line $\mathbb{A}^1$, by considering the logarithm of the intensity modulo some arbitrary constant. But the $\mathbb{E}^2 \times \mathbb{A}^1$-space is quite unlike $\mathbb{E}^3$ as becomes evident when one considers Euclidean rotations about some axis in the image plane. Such rotations make no sense because photon catches and lengths are incommensurable physical quantities. The correct way to proceed is to consider "image space" to be a fiber bundle with base space $\mathbb{E}^2$ and fibers $\mathbb{A}^1$. Permissible transformations do not "mix" fibers, and Euclidean rotations about axes in the image plane are not among them.

This situation is typical in many contexts. The simplest example is perhaps a graph $y = f(x)$, where x and y are incommensurable physical quantities. Although the graph is evidently a curve in the xy-plane, it makes no sense to compute its Euclidean curvature as the result will depend on irrelevant transformations of the y-domain (Fig. 1).

A formal way to deal with such problems is to treat the y-axis as an isotropic dimension. Then the metric in the plane is essentially the separation in the x-dimension, the y-separation being treated as isotropic, i.e., nil. Thus the distance of points $\{x_1, y_1\}$ and $\{x_2, y_2\}$ is taken to be $x_2 - x_1$. Notice that this implies that points $\{x, y_1\}$ and $\{x, y_2\}$ with $y_1 \neq y_2$ are at mutually zero distance, *yet different*. One denotes such points "parallel" and assigns them the "special distance" $y_2 - y_1$. Only parallel points have a special distance, generic points only a proper distance. The group of "isotropic motions":

$$x' = x + t_x, \tag{1}$$

$$y' = \alpha x + y + t_y, \tag{2}$$

conserves proper distance and, in the case of parallel points, special distance. This group is fit to replace the group of Euclidean movements:

$$x' = x \cos\alpha - y \sin\alpha + t_x, \tag{3}$$

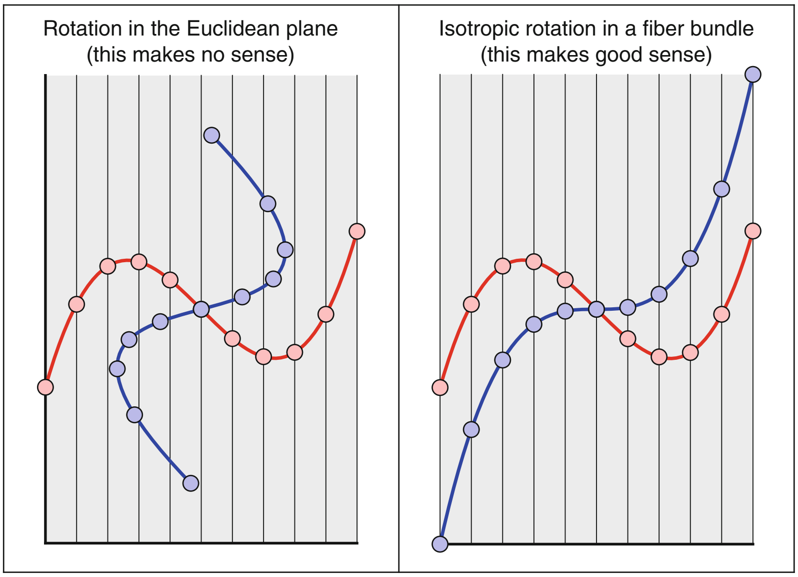
$$y' = x \sin\alpha + y \cos\alpha + t_y, \tag{4}$$

and indeed has a somewhat similar (with important differences!) structure.

One obtains this group if the xy-plane is interpreted as the dual number plane. Dual numbers are complex numbers $z = x + \varepsilon y$, where the imaginary unit ε is defined as a nontrivial (i.e., not equal zero) solution of the quadratic equation $\varepsilon^2 = 0$. Thus $\varepsilon \neq 0$ whereas $\varepsilon^2 = 0$, from which one derives that neither $\varepsilon > 0$, nor $\varepsilon < 0$. Thus one is forced to use intuitionistic logic, for instance dropping the law of the excluded middle. A concrete representation is by way of matrices:

$$z = x + \varepsilon y = \begin{pmatrix} x & y \\ 0 & x \end{pmatrix}, \tag{5}$$

then addition and multiplication of dual numbers may be done by matrix algebra, similar to the conventional complex numbers $x + \mathrm{i}y$ with imaginary unit $\mathrm{i}^2 = -1$,



Isotropic Differential Geometry in Graph Spaces, Fig. 1 At *left* a graph in *red*, and the graph after a Euclidean rotation (*in blue*). This obviously makes no sense at all: the *blue curve* is not even a graph anymore! This is not a fiber bundle. At *right* the *red graph* has been subjected to an isotropic rotation. This makes perfect sense, one obtains another graph. Notice that the points move up and down along the fibers of the fiber space, they never leave their fiber, fibers "do not mix"

which are modeled through matrix algebra with the matrices:

$$z = x + \mathrm{i}y = \begin{pmatrix} x & -y \\ y & x \end{pmatrix}. \qquad (6)$$

However, although perhaps less scary, these matrix models are an unnecessary pain in hand calculations.

A linear transformation $z' = az + b$ with $a = 1 + \varepsilon\alpha$ and $b = t_x + \varepsilon t_y$ becomes:

$$\begin{aligned} z' &= (1 + \varepsilon\alpha)(x + \varepsilon y) + (t_x + \varepsilon t_y) \qquad (7) \\ &= (x + t_x) + \varepsilon(\alpha x + y + t_y), \end{aligned}$$

(using $\varepsilon^2 = 0$), i.e., exactly the transformation given above. Apparently the dual imaginary unit is an "infinitesimal." Indeed, the full Taylor expansion of a function F about x is:

$$F(x + \varepsilon h) = F(x) + \varepsilon h F'(x). \qquad (8)$$

Specifically, one has:

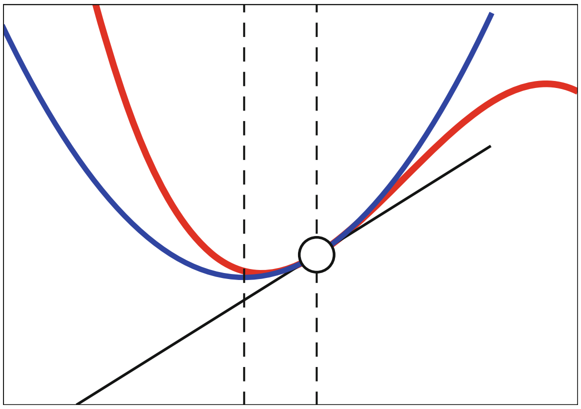
$$\begin{aligned} \sin \varepsilon\xi &= \varepsilon\xi, \qquad (9) \\ \cos \varepsilon\xi &= 1, \qquad (10) \\ \mathrm{e}^{\varepsilon\xi} &= 1 + \varepsilon\xi, \qquad (11) \end{aligned}$$

thus trigonometry becomes really convenient. The polar representation of a dual number becomes:

$$z = x + \varepsilon y = x(1 + \varepsilon \frac{y}{x}) = |z|\mathrm{e}^{\varepsilon \arg z}. \qquad (12)$$

The dual angle is $\arctan y/x = y/x$ (notice that an isotropic angle equals its tangent!), thus the angle measure is parabolic instead of elliptic. Angles do not repeat with period 2π as in the Euclidean plane, but run between $\pm\infty$. Rotating the point 1 (that is $1 + \varepsilon\, 0$) about the origin over an angle α yields $1 + \varepsilon\, \alpha$, thus the line $x = 1$ is (part of) a unit circle. This brings one back to the original construction, and the rotations do not "mix" the x and y dimensions in a way that would be nonsense from the perspective of physics.

The group of proper motions (translations and rotations) of the dual plane leads to a differential geometry of curves that differs from that of the Euclidean plane. Consider the curve $z(x) = x + \varepsilon y(x)$. It is evidently parameterized by arc length, for $|z_x|^2 = 1$. The tangent is $t(x) = z_x = 1 + \varepsilon y_x(x)$, and is a unit vector, for $|t(x)| = 1$. The unit normal is ε, for $t_x(x) = \varepsilon y_{xx}(x)$ with (special) length $y_{xx}(x)$. Thus the normal is constant along the curve and useless for the purposes of differential geometry. The slope of the tangent is well defined though, the tangent subtends an angle $y_x(x)$ with the x-axis. The derivative of this angle with arclength is $y_{xx}(x)$, thus one concludes that the curve has



Isotropic Differential Geometry in Graph Spaces, Fig. 2 Consider the local Taylor expansion of the *red curve* at the point indicated by the *white dot*. The first order is the drawn *black line*, the second-order approximation is the *blue parabola*. The center of this "circle of the second kind" is indicated by the leftmost *dotted line*. The other *dotted line* is the normal direction

curvature $\kappa(x) = y_{xx}(x)$. Notice that it is a much simpler expression than one has in the Euclidean plane, which is:

$$k(x) = \frac{y_{xx}(x)}{(1 + y_x(x)^2)^{3/2}}. \qquad (13)$$

As expected, the Euclidean and the dual curvatures agree to first order, and for very shallow curves (infinitesimally near the x-axis) the Euclidean curvature degenerates to the dual curvature.

A curve:

$$z(x) = \varepsilon \frac{(x-c)^2}{2R}, \qquad (14)$$

has curvature $1/R$, thus a radius of curvature R and is centered on $x = c$. It is evidently a circle in some sense, though different from the circle encountered above. One denotes $x = \pm 1$ a unit circle of the first kind, with center at the origin, $\varepsilon x^2/2$ a unit circle of the second kind, centered at the origin. The local second-order Taylor expansion of a curve is illustrated in Fig. 2. It is a parabola with isotropic axis, thus a "circle of the second kind." It is the osculating circle to the curve in the isotropic geometry. The radius of the osculating circle is evidently the reciprocal of the second derivative, thus a curve $x + \varepsilon y(x)$ has curvature $y_{xx}(x)$ as argued above.

The differential geometry of curves and surfaces in a fiber bundle $\mathbb{E}^2 \times \mathbb{A}^1$ can be handled in a similar manner. All expressions are much simpler than those in Euclidean differential geometry, which is a very useful property, apart from the advantage that they make sense for a change. (Inappropriate applications of expressions taken from Euclidean differential geometry occur very frequently in computer vision and image processing. Although they certainly yield numerical results, they strictly speaking make no sense.) Thus the mean curvature of a surface $\{x, y, z(x, y)\}$ in Monge form becomes $2H = z_{xx} + z_{yy}$, the Gaussian curvature $K = z_{xx}z_{yy} - z_{xy}^2$, and so forth. Like in the planar case discussed above, the normal is constant, and thus useless. One uses the spatial attitude of the tangent plane instead. Any point of the surface may be mapped on the unit sphere of the second kind $z(x, y) = (x^2 + y^2)/2$ through parallelity of tangent planes. Even more conveniently, one notices that the xy-plane $\{x, y, 0\}$ is the stereographic projection of this sphere, thus conformal, but most remarkably – because different from the Euclidean case – also isometric. Thus the Gauss map becomes

$$\{x, y, z(x, y)\} \in \mathbb{E}^2 \times \mathbb{A}^1 \mapsto \{-z_x(x, y), \\ - z_y(x, y)\} \in \mathbb{R}^2, \qquad (15)$$

a map that is familiar in computer vision as "gradient space." Gradient space is often used by way of a "linear approximation," but it is really the exact Gauss map (or "spherical image") in terms of the appropriate differential geometry.

The geometry of single isotropic space is well understood, although almost all of the literature is in German. The paper by Pottmann is the only reference in English on the general (space) setting, the book by Yaglom (translated into English from Russian) is an excellent introduction to the geometry of the dual plane.

Open Problems

This section introduced a very simple setting. In general one may have to deal with a graph over a curved surface. The paper by Pottmann gives some leads as how to handle such more general cases.

References

1. Koenderink JJ, van Doorn AJ (2002) Image processing done right. In: Proceedings of the European conference on computer vision (ECCV 2002). Lecture notes in computer science, vol 2350. Springer, Heidelberg, pp 158–172
2. Pottmann H, Opitz K (1994) Curvature analysis and visualization for functions defined on Euclidean spaces or surfaces, Comput Aided Geom Des 11: 655–674
3. Yaglom IM (1979) A simple non–Euclidean geometry and its physical basis: an elementary account of Galilean geometry and the Galilean principle of relativity. (trans: Shenitzer A). Springer, New York (translated from Russian)

Iterative Closest Point (ICP)

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Synonyms

ICP

Definition

Iterative closest point (ICP) is a popular algorithm employed to register two sets of curves, two sets of surfaces, or two clouds of points.

Background

The ICP technique was proposed independently by Besl and McKay [1] and Zhang [2] in two different contexts. Besl and McKay [1] developed the ICP algorithm to register partially sensed data from rigid objects with an ideal geometric model, prior to shape inspection. So this is a subset–set matching problem because each sensed point has a correspondence in the ideal model. Zhang [2] developed the ICP algorithm in the context of autonomous vehicle navigation in rugged terrain based on vision. His algorithm is used to register a sequence of sensed data in order to build a complete model of the scene and to plan a free path for navigation. So this is a subset–subset matching problem because a fraction of data in one set does not have any correspondence in the other set. To address this issue, Zhang's ICP algorithm has integrated a statistical method based on the distance distribution to deal with outliers, occlusion, appearance, and disappearance. However, both algorithms share the same idea: iteratively match points in one set to the closest points in another set and refine the transformation between the two sets, with the goal of minimizing the distance between the two sets of point clouds.

Theory

The ICP algorithm is very simple and can be summarized as follows:

- *Input:* two point sets, initial estimation of the transformation
- *Output:* optimal transformation between the two point sets
- *Procedure:* Iterate the following steps:
  (i) Apply the current estimate of the transformation to the first set of points.
  (ii) Find the closest point in the second set for each point in the first transformed point set.
  (iii) Update the point matches by discarding outliers.
  (iv) Compute the transformation using the updated point matches, until convergence of the estimated transformation.

Here are a few comments on this general algorithm:

- Depending on the nature of the point sets, various pose estimation techniques described in the earlier sections can be used to compute the transformation between the two sets.
- The step of finding the closest point to a given point is generally the most time-expensive one. However, this step can be easily parallelized.
- Many data structures can be used to accelerate the finding of the closest point. They include k-D tree and octree.
- Instead of using all points from the first set, a selected subset of points (such as high curvature points) can be used to speed up the process, with only moderate sacrifice of the final accuracy.
- The above algorithm is not symmetric. Let point $\hat{p}_i'$ in the second set be the closest point to a point p_i in the first set. In the other direction, point p_i is, in general, not necessarily the closest point to $\hat{p}_i'$. In

order to make the algorithm symmetric, we can find the closest point in the first transformed point set for each point in the second set and add these point matches to the overall set of matches. Better results can then be obtained at the expense of additional computational cost.

- When the ICP algorithm is applied to register curves or surfaces, they need to be sampled. The final accuracy depends on the density of sampling. The denser the sampling is, the higher the registration quality will be, but the more the computation will be required.

For more detailed and extensive discussions on ICP, the interested reader is referred to Sects. 7 and 8 of Zhang's paper [2].

There are several variants to the ICP algorithm. A useful variation is to substitute the point-to-point distance with point-to-plane distance [3]. The point-to-plane distance allows one surface to slide tangentially along the other surface, making it less likely get stuck in local minima. Consider a point p_i in the first set. Let point $\hat{p}_i'$ in the second set be its closest point. Let the surface normal at point p_i be n_i (a unit vector). Then, the point-to-plane distance measure is given by

$$d_i = n_i^T(\hat{p}_i' - p_i).$$

Surface normals can be precomputed to save computation.

References

1. Besl PJ, McKay ND (1992) A method for registration of 3-d shapes. IEEE Trans Pattern Anal Mach Intell 14(2):239–256
2. Zhang Z (1994) Iterative point matching for registration of free-form curves and surfaces. Int J Comput Vis 13(2):119–152. Also Research Report No.1658, INRIA Sophia-Antipolis, 1992
3. Chen Y, Medioni G (1992) Object modelling by registration of multiple range images. Image Vis Comput 10(3):145–155

Ives Transform

►von Kries Hypothesis