V

Variable Selection

►Feature Selection

Variational Analysis

Bastian Goldluecke
Department of Computer Science, Technische
universität München, München, Germany

Synonyms

Calculus of variations

Related Concepts

►Total Variation; ►Variational Method

Definition

In mathematics, the term *variational analysis* usually denotes the combination and extension of methods from *convex optimization* and the classical *calculus of variations* to a more general theory [5]. However, in computer vision literature, the term is frequently encountered as just a synonym for *calculus of variations*. This branch of mathematics deals with the minimization of functionals, which are real-valued functions on infinite-dimensional spaces, most frequently spaces of functions.

Background

In the continuous world view, images are modeled as functions on a domain $\Omega \subset \mathbb{R}^n$. Geometric entities like curves and surfaces are manifolds, which can be described as level sets of functions or by the characteristic functions of their interior region. Consequently, computer vision problems can often successfully be formulated as minimization problems on infinite-dimensional spaces, where the solution is given as the minimizer of an *energy functional* E: $\mathcal{V} \to \mathbb{R}$ on the space $\mathcal{V}$ of admissible functions. The minimization of such a functional thus requires a calculus on infinite-dimensional spaces, which is provided by the classical branch of mathematics called *calculus of variations* [3, 4]. It provides necessary conditions which have to be satisfied by the minimizing function, probably most well known is the *Euler-Lagrange equation*, a partial differential equation for the unknown. However, the methods of the calculus of variations usually provide only conditions for local minima, while it is of course desirable to formulate models where one has some guarantees about the optimality of the solution.

Recently, there has therefore been much effort to formulate computer vision in a way that the final minimization problem is a *convex functional*. The main advantage is that there are no local minima in the sense that each local minimizer is automatically a global optimum. Furthermore, for optimization, it is possible to enhance the techniques from the calculus of variations with methods from *convex optimization*, thus obtaining very efficient minimization algorithms [1, 4, 5].

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

The formulation of variational models as convex problems requires convex regularizers. A particularly powerful regularizer was found to be the *total variation* of a function [2]. Besides convexity, it has certain interesting geometric properties which make it possible to obtain convex formulation of segmentation and minimal surface problems. See the entry on *total variation* for more details.

Theory

See the entry on the *variational method*.

References

1. Attouch H, Buttazzo G, Michaille G (2006) Variational analysis in Sobolev and BV spaces. MPS-SIAM series on optimization. Society for Industrial and Applied Mathematics, Philadelphia
2. Chambolle A, Caselles V, Cremers D, Novaga M, Pock T (2010) An introduction to total variation for image analysis. Radon Ser Comput Appl Math 9:263–340
3. Gelfand IM, Fomin SV (2003) Calculus of variations. Dover publications reprint of the 1963 edn. Dover Publications Inc., Mineola, NY
4. Luenberger D (1969) Optimization by vector space methods. Wiley, New York
5. Rockafellar RT, Wets R (2005) Variational analysis. Springer, New York

Variational Method

Bastian Goldluecke
Department of Computer Science, Technische universität München, München, Germany

Related Concepts

►Total Variation; ►Variational Analysis

Definition

The variational method is a way to solve problems given in the form of a *variational model*, i.e., as an energy minimization problem on an infinite-dimensional space which is typically a function space. It employs tools from the mathematical framework of *variational analysis*.

Background

Ikeuchi and Horn's shape-from-shading paper and Horn and Schunck's optical flow paper, which appeared in AIJ simultaneously, are the earliest representative works to introduce a variational method to computer vision [7]. Inspired by the extraordinary success of the idea, variational methods have been extensively studied in computer vision and become a very popular tool for a wide variety of problems. They are particularly successful in mathematical image processing, where they are used to describe fundamental low-level problems, like image segmentation [9, 11], denoising [15], and deblurring [3], but have also been employed for high-level tasks like 3D reconstruction [4, 10].

In a variational model, the solution to a problem is obtained as the minimizer of an energy defined on a typically infinite-dimensional space, for example, a function space. The underlying world view is a continuous one similar to classical physics, where images are regarded as functions on $\mathbb{R}^n$, while curves and surfaces, encountered, for instance, in image segmentation and 3D reconstruction, are regarded as manifolds.

In view of this, it is not surprising that the mathematical framework of *variational analysis* used to deal with these problems was developed in close conjunction with new insights in physics in the early twentieth century [5]. It is a calculus for functionals on infinite-dimensional spaces, which together with tools from differential geometry and measure theory yields the Euler-Lagrange equations of such a functional as a necessary condition for a minimum. Commonly, solutions to the Euler-Lagrange equations are obtained either by directly solving a discretization or via a gradient descent technique.

Usually, only a local minimum is obtained this way. Recently, however, much effort has been devoted to formulate variational problems with convex energies, either directly or using relaxation techniques. In the case of a convex energy, the solution to the Euler-Lagrange equation yields a global optimum. Furthermore, the methods from variational analysis can be enhanced with methods from convex analysis and convex optimization, yielding powerful and efficient optimization methods.

Theory

There are two classical types of variational problems, which require a slightly different mathematical apparatus. The first class is concerned with the minimizer of a functional which is defined on a (usually infinite-dimensional) vector space $\mathcal{V}$, for example, a function space. It typically arises when the goal is to recover images, as in the problems of image denoising and deblurring.

The second type is more complex and deals with weighted minimal surfaces, i.e., minimizers of a class of functionals defined for surfaces. This type of problem arises when one tries to recover manifolds, like curves in image segmentation or surfaces in 3D reconstruction. It can be related to the first case by introducing a variation of a surface and employing methods from differential geometry.

Functionals on vector spaces The foundation for the variational method for functionals on vector spaces is the *variational principle*, which states that a minimum of a functional is a *stationary point* [2, 5]. Let $E : \mathcal{V} \to \mathbb{R}$ be a functional on the Banach space $\mathcal{V}$ and then a point $u \in \mathcal{V}$ is called stationary if the *Gâteaux derivatives* $\delta E(u; h)$ in all directions h vanish, i.e.,

$$0 = \delta E(u; h) := \lim_{\alpha \to 0} \frac{1}{\alpha} \left(E(u + \alpha h) - E(u) \right). \quad (1)$$

Typically, the functional E is given as an integral over a domain Ω. In order to apply the variational principle in this case, one writes the Gâteaux differential in the form

$$\delta E(u; h) = \int_{\Omega} \phi_u \cdot h \, \mathrm{d}x, \quad (2)$$

where ϕ_u is a function on Ω which depends on u.

The *du Bois-Reymond Lemma* then implies that if $\delta E(u; h) = 0$ for all h, then in fact $\phi_u = 0$ on Ω. This is a partial differential equation for the unknown u and a necessary condition which has to be satisfied by a minimizer. It is called the *Euler-Lagrange equation* of the functional E.

A frequent form for the energy E is the formulation

$$E(u) = \int_{\Omega} \mathcal{L}(u(x), \nabla u(x), x) \, \mathrm{d}x, \quad (3)$$

with a *Lagrangian* function $\mathcal{L}$. In this case, an explicit formula for the Euler-Lagrange equation can be derived using the chain rule and theorem of Gauss for integration by parts. It reads

$$\frac{\partial \mathcal{L}}{\partial u} - \operatorname{div} \left(\frac{\partial \mathcal{L}}{\partial (\nabla u)} \right) = 0, \quad (4)$$

with either Neumann boundary conditions on u or Dirichlet boundary conditions on possible directions h.

Minimal surfaces. A minimal surface $\Sigma \subset \mathbb{R}^n$ is a local minimizer of the surface area integral

$$\mathcal{A}(\Sigma) := \int_{\Sigma} \Phi \, dA, \quad (5)$$

with a real-valued weight function $\Phi \geq 0$ and dA denoting the surface area element. The dimension of the surface is one less than the embedding space. Problems of this form can be handled by introducing a *variation of the surface* as a differentiable map

$$X : \Sigma \times (-\epsilon, \epsilon) \to \mathbb{R}^n, \quad (6)$$

where $X(\Sigma, 0) = \Sigma$ and a one-parameter family $\Sigma_\tau = X(\Sigma, \tau)$ of regular surfaces is given depending on $\tau \in \mathbb{R}$. The variational principle then implies that a minimal surface satisfies

$$\left. \frac{d}{d\tau} \right|_{\tau=0} \mathcal{A}(\Sigma_\tau) = 0 \quad (7)$$

for all possible variations. In [6], the Euler-Lagrange equation was derived in arbitrary dimension using methods from Cartan geometry [16], for the general case that Φ depends on the location s on the surface and the local normal $\mathbf{n}$. For a more elementary deduction for two-dimensional surfaces, see, for example, [4, 14], where [14] also deals with other classes of minimal surfaces which minimize weighted mean or Gaussian curvature.

The main result of [6] is that a minimal surface necessarily satisfies

$$(\Phi_s, \mathbf{n}) - \operatorname{Tr}(\mathbf{S}) \, \Phi + \operatorname{div}_{\Sigma}(\Phi_{\mathbf{n}}) = 0, \quad (8)$$

where $\operatorname{Tr}(\mathbf{S})$ is the trace of the shape operator of the surface, also known as the Weingarten map or second fundamental tensor. Using either explicit surface evolution methods or the *level set method* [13], a local minimum can be obtained via gradient descent as a stationary solution of a corresponding evolution equation.

In a more modern framework, the weighted surface area is replaced with the *weighted total variation* [1, 2] of the characteristic function of the surface interior. Optimization then takes place over the set of binary functions of bounded variation, which means that the difficult minimal surface problem has been reduced to the more simple first case discussed above. Using convex relaxation techniques, this allows to solve certain classes of weighted minimal surface problems, which, for instance, arise in segmentation [12] and 3D reconstruction [10], in a globally optimal manner. The entry on *total variation* provides more details on this topic.

References

1. Ambrosio L, Fusco N, Pallara D (2000) Functions of bounded variation and free discontinuity problems. Oxford University Press, Oxford/New York
2. Attouch H, Buttazzo G, Michaille G (2006) Variational analysis in Sobolev and BV spaces. MPS-SIAM series on optimization. Society for Industrial and Applied Mathematics, Philadelphia
3. Chan T, Wong C (1998) Total variation blind deconvolution. IEEE Trans Image Process 7(3):370–375
4. Faugeras O, Keriven R (1998) Variational principles, surface evolution, PDE's, level set methods, and the stereo problem. IEEE Trans Image Process 7(3):336–344
5. Gelfand IM, Fomin SV (2003) Calculus of variations. Dover publications reprint of the 1963 edn. Dover Publications Inc., Mineola, NY
6. Goldluecke B, Ihrke I, Linz C, Magnor M (2007) Weighted minimal hypersurface reconstruction. IEEE Trans Pattern Anal Mach Intell 29(7):1194–1208
7. Ikeuchi K, Horn BKP (1981) Numerical shape from shading and occluding boundaries. Artif Intell 17: 141–184
8. Horn B, Schunck B (1981) Determining optical flow. Artif Intell 17:185–203
9. Kass M, Witkin A, Terzopoulos D (1987) Snakes: active contour models. Int J Comput Vis 1:321–331
10. Kolev K, Klodt M, Brox T, Cremers D (2009) Continuous global optimization in multiview 3D reconstruction. Int J Comput Vis 84(1):80–96
11. Mumford D, Shah J (1989) Optimal approximation by piecewise smooth functions and associated variational problems. Commun Pure Appl Math 42:577–685
12. Nikolova M, Esedoglu S, Chan T (2006) Algorithms for finding global minimizers of image segmentation and denoising models. SIAM J Appl Math 66(5): 1632–1648
13. Osher S, Fedkiw R (2003) Level set methods and dynamic implicit surfaces. Volume 153 of applied mathematical sciences. Springer, New York
14. Overgaard N, Solem J (2007) The variational origin of motion by gaussian curvature. In: Proceedings of the 1st international conference on Scale space and variational methods in computer vision (SSVM)
15. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. Physica D 60(1–4):259–268
16. Sharpe RW (1997) Differential geometry. Volume 166 of graduate texts in mathematics. Springer, New York

Variational Methods

▶Geodesics, Distance Maps, and Curve Evolution

Veiling Glare

▶Lens Flare and Lens Glare

Velvety Reflectance

▶Asperity Scattering

Video Alignment and Stitching

▶Video Mosaicing

Video Mosaicing

Zhigang Zhu
Computer Science Department, The City College of New York, New York, NY, USA

Synonyms

Panoramic image generation; Video alignment and stitching; Video mosaicking

Related Concepts

▶Image Registration

Definition

Image mosaicing is the process of generating a composite image (mosaic) from a video sequence, or in general from a set of overlapping images of a scene or an object, usually resulting in a mosaic image with a larger field of view than any of the original images.

Background

When collecting video of a scene or object, each individual image in the video may be limited compared to the desired final product, including limitations in the field of view, dynamic range, or image resolution. This is the case not only with personal video capture [1, 9, 13] but also with image-based rendering [11, 14, 15], aerial videography [7, 10, 18–20], and document digitization [5]. Generating mosaics with larger fields of view [5, 6, 9, 13, 14, 19], higher dynamic ranges [4], and/or higher image resolutions [8] facilitates video viewing, video understanding, video transmission, and archiving. When the major objective of video mosaicing is to generate a complete (e.g., 360°) view of an object (or a scene) by aligning and blending a set of overlapping images, the resulting image is also called a video panorama [9, 14, 15].

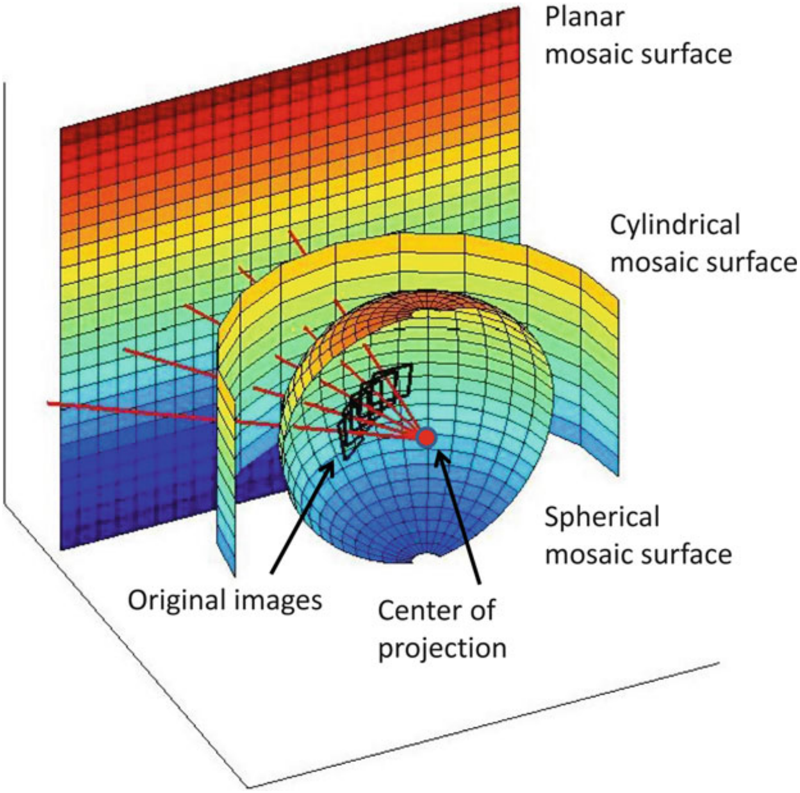
Theory and Application

Video mosaicing takes in a video sequence and generates one or more mosaiced images with either a larger field of view, a higher dynamic range, a higher image resolution, or a combination of them. This entry will mainly discuss the principles in generating large field of view mosaics (panoramas), but the principles can also be (mostly) applied to mosaics for other objectives (high dynamic range imaging and super-resolution imaging). Here, *video* mosaicing implies that the images in the sequence are taken by a video camera, usually at 30 frames per second, but images taken by a digital camera such that there is a large amount of spatial overlap between two consecutive frames can also be viewed as a video sequence.

There are three key components in a typical video mosaicing algorithm: motion modeling, image alignment, and image composition. Depending on the type of camera motion and the structure of the objects or scenes, the *motion model* can be a 2D rigid motion model (rotation, translation, scaling), an affine model, a perspective model (homography), or a full 3D motion model.

Many popular video mosaicing methods [16], for example, in [4, 15], assume a pure rotation model of the camera in which the camera rotates around its center of projection (i.e., the optical center, sometimes called nodal point). In this case, the motion between two consecutive frames can be modeled by a homography, which is a 3×3 matrix. Then, depending on the fields of view (FOVs) of the mosaic, the projection model of the mosaic can be a perspective projection (FOV is less than 180°), a cylindrical projection (FOV is 360° in one direction), or a spherical projection (full 360° FOV in both directions). Figure 1 illustrates the relations between the original images and the three types of mapping surfaces each image can be projected onto: planar, cylindrical, and spherical.

However, the applications of video mosaics from a pure rotation camera are limited to mostly consumer applications such as personal photography, entertainment, and online maps. For more specialized applications such as surveillance, remote sensing, robot navigation, and land planning, to name a few, the motion of the camera cannot be limited to a pure rotation. Translational motion usually cannot be avoided, causing the *motion parallax* problem to arise. There are three kinds of treatments for the motion parallax problem. First, when the translational components are relatively small, the motion models can be approximated by a pure rotation. In this case, the generated mosaics lack geometric accuracy, but with some treatments for the small motion parallax and moving targets, such as de-ghosting [15], the mosaics generally look very good. Second, if the scene can be regarded as planar, for example, because the distance between the camera and the scene is much larger than the depth range of the scene, the perspective motion model (homography) or, in some applications, a 2D rigid motion model or an affine model can be used [6, 10, 20]. In these cases, the problems are much simpler due to the 2D scene assumption. Finally, a



Video Mosaicing, Fig. 1 Mapping a set of overlapping images into a mosaic: planar, cylindrical, or spherical

3D camera motion model is applied when the translational components of the camera motion are large and the scene is truly 3D. In this case, motion parallax cannot be ignored or eliminated. Examples include a camera mounted on an airplane or a ground vehicle translating a large distance [7, 11, 13, 19], or a camera's optical center moving on a circular path [9, 14]. Here, multi-perspective projection models are used to generate the mosaics, enabling stereo mosaics or stereo panoramas to be created that preserve the 3D information in the scene, allowing the structure to be reconstructed and viewed in 3D. In this case, the accuracy of geometric modeling and image alignment is crucial for achieving the accuracy of 3D reconstruction and viewing.

Image alignment (or *image registration*) is the process of finding the alignment parameters (e.g., the homography in the rotational case) between two consecutive images. Image alignment is a critical step in mosaic generation, for both seamless mosaicing and for accurate geometric representation. There are two approaches to image registration: direct methods or feature-based methods. In a direct method, a correlation approach is used to find the motion parameters. Here, the images are divided into small blocks and each block in the first image is searched for over a predefined spatial range in the second image. The best match is determined by finding the maximal correlation value. Other approaches such as using optical flow or using an iterative optimization framework also belong to the direct methods, in which no explicit feature points are extracted. In a feature-based method, a feature detection operator such as the Harris corner or SIFT (Scale Invariant Feature Transform) detector is used first, then the detected features are matched over the two frames to build up matches [16]. Either way, a parameter model is fitted using all the matches, usually using a robust parameter estimation method to eliminate erroneous feature matches. For more accurate or consistent results, a global optimization can be applied among more than two frames. For example, global alignment may be applied to all the frames in a full 360° circle in order to avoid gaps between the first and the last frame [15].

Image composition is the step of combining aligned images together to form the viewable mosaic. There are three important issues in this step: compositing surface determination, coordinate transformation and image sampling, and pixel selection and blending. Mosaicing with the rotational camera model is a good starting point to discuss these issues (Fig. 1); mosaic compositing under other motion models are discussed afterwards.

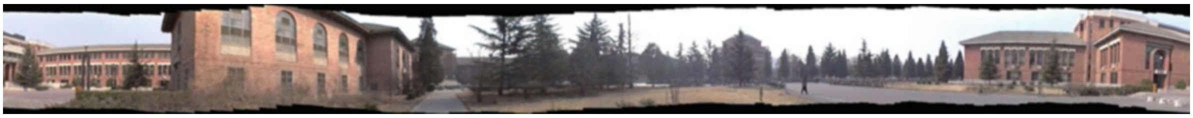
If the video sequence only has a few images, then one of the images can be selected as the reference image, and all the other images are warped and aligned with this reference image. In this case, the reference image with a perspective projection is the compositing surface, and therefore the final mosaic is a larger perspective image, which is an extension of the field of view of the reference image. However, this approach only works when the view angles of the images span less than 90°. If the camera rotates more than 90°, a cylindrical or a spherical surface should be selected as the compositing surface. A cylindrical surface is a good representation when a full 360 panoramic mosaic is to be generated, in one direction. And a spherical surface is suitable if 360° × 360° mosaics are to be created.

After a compositing surface is selected, the next issue is coordinate transformation and sampling. This is also called image warping. Given the motion parameters obtained in the image registration step, the mapping between each frame to the final compositing surface can be calculated: For any pixel in an original image frame, its pixel location in the compositing surface can be calculated. For generating dense pixels, an interpolation schema is needed, such as nearest neighbor, bilinear, or cubic interpolation methods. Usually a backward mapping relation is utilized such that in the mapping area on the compositing surface, each pixel obtains a value from the original image frame, line by line, and column by column. Therefore, for each integer pixel location in the mosaic, a decimal pixel location can be found in the original image; then an interpolation method is used in the original image to generate the value of the pixel in the mosaic.

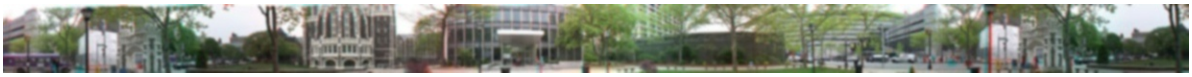
The third important issue in image composition is pixel selection and blending. Naturally in generating mosaics, there are overlaps among consecutive frames, resulting in two key questions: First, *where do we place the seam (i.e., the stitching line)* (the pixel selection problem)? Second, *how do we select the values of pixels in the overlapping areas* (the pixel blending problem)? For the second problem, the simplest methods are to average all the pixels in the same location in the overlapping area, or to use their median value. The former might create a so-called ghost effect due to moving objects, small motion parallax, or illumination changes, while the latter approach may generate a slightly better view effect. More sophisticated blending methods include Laplacian pyramid blending [3] and gradient domain blending [1]. The pixel selection problem is important when moving objects or motion parallax exists in the scene. In these cases, to avoid a person being cut in half or appearing twice in the mosaic, or to avoid cutting a 3D object that exhibits obvious motion parallax and hence could produce obvious misalignment in the mosaic, an optimal seam line can be selected at pixel locations where there are minimum misalignments between two frames [4].

Other considerations in image composition are high dynamic range imaging [4] and improved image resolution mosaicing [8]. For the former, a composite mosaic represents larger dynamic ranges than individual frames using varying shutter speeds and exposures, while the latter uses the camera motion to generate higher spatial resolution in the mosaiced image than that of the original images.

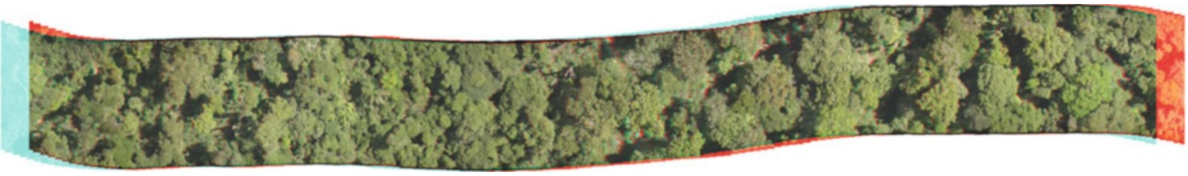
So far the discussions on image composition have focused primarily on 2-D mosaics, assuming either the camera motion is (almost) a pure rotation, or the scene is flat or very far from the camera, in order to avoid or reduce the motion parallax problem. When motion parallax cannot be avoided, 3-D mosaics have to be considered. Methods have been proposed to generate mosaics, for example, for curved documents based on 3-D reconstruction [5], when the camera motion has translational components. Needless to say, with 3-D reconstruction, a composite image with a new perspective view, or a new projection representation (such as orthogonal projection), can be synthesized from the original images. However, the drawback of this approach is a full 3-D reconstruction is needed, which is both computationally expensive and prone to noise. A more practical yet still fundamental approach without 3-D reconstruction is to generate multi-perspective mosaics from a video sequence, such as mosaics on an adaptive manifold [10], creating stitched images of scenes with parallax [7], and creating multiple-center-of-projection images [11]. When the dominant motion of the camera is translation, the projection model of the mosaic can be a parallel-perspective projection, in that the projection in the direction of the motion is parallel,



Video Mosaicing, Fig. 2 A 360° panoramic mosaic generated on a cylindrical surface http://www-cs.engr.ccny.cuny.edu/~zhu/ThlibCylinder.JPG



Video Mosaicing, Fig. 3 A pair of concentric mosaics of the City College of New York campus http://www-cs.engr.ccny.cuny.edu/~zhu/CSCI6716/CCNYCampus.jpg



Video Mosaicing, Fig. 4 A pair of pushbroom mosaics of the Amazon rain forest http://www-cs.engr.ccny.cuny.edu/~zhu/57z10StereoColor.jpg

whereas the projection perpendicular to the motion remains perspective. This kind of mosaic is also called pushbroom mosaic [17] since the projection model of the mosaic in principle is the same as pushbroom imaging in remote sensing. A more interesting case is that by selecting different parts of individual frames, a pair of stereo mosaics can be generated that exhibit motion parallax, while each of them represent a particular viewing angle of parallel projection [19]. To generate stereo mosaics, the motion model is 3D, and therefore, a bundle adjustment for 3D camera orientation is needed. The projection model is parallel perspective, and therefore, the composting surface is a plane that holds the parallel-perspective image. To generate a true parallel-perspective view in each mosaic for accurate 3D reconstruction, pixel selection is carried out for that particular viewing angle and a coordinate transformation is performed based on matches between at least two original images for each pixel. A similar principle can be applied to concentric mosaics with circular projection [9, 14].

In some applications such as surveillance and mapping, geo-referencing mosaicing is also an important topic. This is usually done when geo-location metadata is available, for example, from GPS and IMU measurements [18, 20] taken with the video/images. Geo-referenced mosaics assign a geo-location to each pixel either by directly using the metadata from the video frames used to generate the mosaic or, when metadata is not available, by aligning the video frames to a geo-referenced reference image such as a satellite image.

Video mosaicing techniques are also used for dynamic scenes, such as to generate dynamic pushbroom mosaics for moving target detection [17] and to create animated panoramic video textures in which different portions of a panoramic scene are animated with independently moving video loops [2, 12].

Open Problems

Some open problems can be found in a good survey paper on image alignment and stitching [16]. These include robust alignments for stereo mosaics (or mosaics with motion parallax), mosaics for high dynamic range imaging and for super-resolution imaging, and dynamic mosaics.

Experimental Results

Figure 2 shows a 360° panoramic mosaic represented on a cylindrical surface, which is generated from a video sequence taken by a video camera that roughly rotates around its optical center. Figures 3 and 4 show two stereo mosaics that can be viewed with a pair of 3D glasses, red for the right eye and the cyan for the left eye. High-resolution mosaics can be viewed by clicking the images in the figures in the online edition. The concentric stereo mosaic in Fig. 3 is generated from a video sequence taken by a handheld video camera that undertakes an off-center rotation with 360 degrees of field of view coverage. Figure 4 is a pair of pushbroom stereo mosaics created from a video sequence taken by a camera looking down from an airplane flying over the Amazon rain forest.

References

1. Agarwala A, Dontcheva M, Agrawala M, Drucker S, Colburn A, Curless B, Salesin D, Cohen M (2004) Interactive digital photomontage. ACM Trans Graph 23(3): 292–300
2. Agarwala A, Zheng C, Pal C, Agrawala M, Cohen M, Curless B, Salesin D, Szeliski R (2005) Panoramic video textures. ACM Trans Graph 24(3):821–827
3. Burt PJ, Adelson EH (1983) A multiresolution spline with applications to image mosaics. ACM Trans Graph 2(4): 217–236
4. 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'2006). IEEE Computer Society, Los Alamitos, pp 2498–2505
5. Iketani A, Sato T, Ikeda S, Kanbara M, Nakajima N, Yokoya N (2006) Video mosaicing for curved documents based on structure from motion. In: ICPR, Hong Kong, vol 4, pp 391–396
6. Irani M, Anandan P, Hsu SC (1995) Mosaic based representations of video sequences and their applications. In: ICCV. IEEE Computer Society, Los Alamitos, pp 605–611
7. Kumar R, Anandan P, Irani M, Bergen J, Hanna K (1995) Representation of scenes from collections of images. In: IEEE workshop on representations of visual scenes. IEEE Computer Society, Los Alamitos, pp 10–17
8. Marzotto R, Fusiello A, Murino V (2004) High resolution video mosaicing with global alignment. In: CVPR, vol 1. IEEE Computer Society, Los Alamitos, pp 692–698
9. Peleg S, Ben-Ezra M (1999) Stereo panorama with a single camera. In: IEEE conference on computer vision pattern recognition (CVPR). IEEE Computer Society, Los Alamitos, pp 1395–1401
10. Peleg S, Rousso B, Rav-Acha A, Zomet A (2000) Mosaicing on adaptive manifolds. IEEE Trans Pattern Anal Mach Intell 22:1144–1154
11. Rademacher P, Bishop G (1998) Multiple-center-of-projection images. In: Computer graphics proceedings. Annual conference series. Association for Computing Machinery, New York, pp 199–206
12. Rav-Acha A, Pritch Y, Lischinski D, Peleg S (2005) Dynamosaics: video mosaics with non-chronological time. In: IEEE computer society conference on computer vision and pattern recognition (CVPR'2005). IEEE Computer Society, Los Alamitos, pp 58–65
13. Rousso B, Peleg S, Finci I, Rav-Acha A (1998) Universal mosaicing using pipe projection. In: ICCV. Narosa Publishing House, New Delhi, pp 945–952
14. Shum H-Y, Szeliski R (1999) Stereo reconstruction from multiperspective panoramas. In: Seventh international conference on computer vision (ICCV'99), IEEE Computer Society, Los Alamitos, pp 14–21
15. Shum H, Szeliski R (2000) Systems and experiment paper: construction of panoramic image mosaics with global and local alignment. Int J Comput Vis 36:101–130
16. Szeliski R (2006) Image alignment and stitching: a tutorial. Found Trends Comput Graph Vis 2(1):1–104
17. Tang H, Zhu Z, Wolberg G (2006) Dynamic 3D urban scene modeling using multiple pushbroom mosaics. In: 3DPVT, Chapel Hill, USA, pp 456–463
18. Taylor CN, Andersen ED (2008) An automatic system for creating geo-referenced mosaics from MAV video. In IROS. IEEE, Piscataway, pp 1248–1253
19. Zhu Z, Hanson AR, Riseman EM (2004) Generalized parallel-perspective stereo mosaics from airborne video. IEEE Trans Pattern Anal Mach Intell 26: 226–237
20. Zhu Z, Riseman EM, Hanson AR, Schultz HJ (2005) An efficient method for geo-referenced video mosaicing for environmental monitoring. Mach Vis Appl 16: 203–216

Video Mosaicking

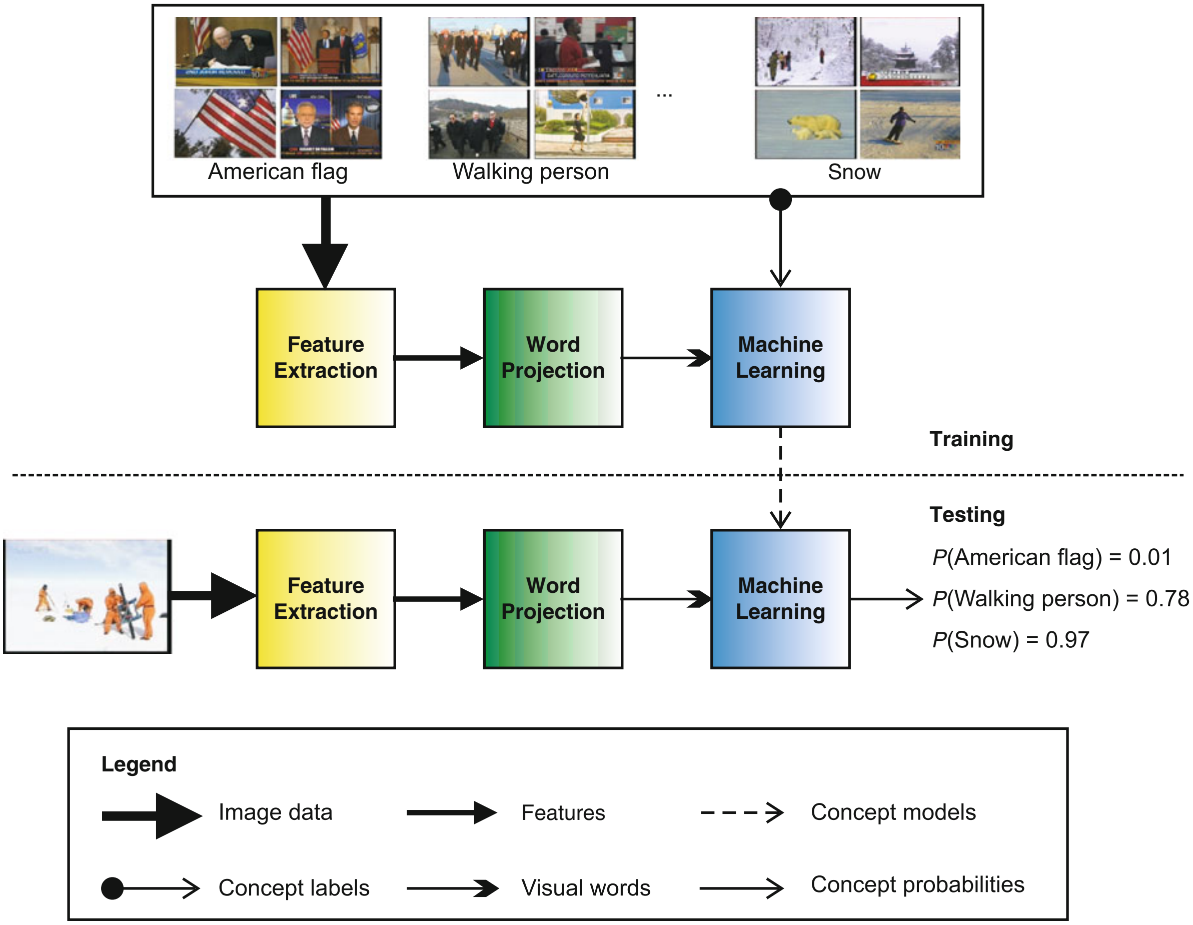
►Video Mosaicing

Video Retrieval

Cees G. M. Snoek[1,3] and Arnold W. M. Smeulders[2,3]
[1]University of Amsterdam, Amsterdam,
The Netherlands
[2]Centre for Mathematics and Computer Science
(CWI), University of Amsterdam, Amsterdam,
The Netherlands
[3]Intelligent Systems Lab Amsterdam, Informatics
Institute University of Amsterdam, Amsterdam,
The Netherlands

Synonyms

Multimedia retrieval; Video search



Video Retrieval, Fig. 1 General scheme for detecting visual concepts in images, with three typical concepts highlighted. First, researchers project extracted image features into visual words. Then they train concept models from both the visual words and the concept labels using machine learning. Finally, during testing, researchers assign concept probabilities to previously unlabeled images

Definition

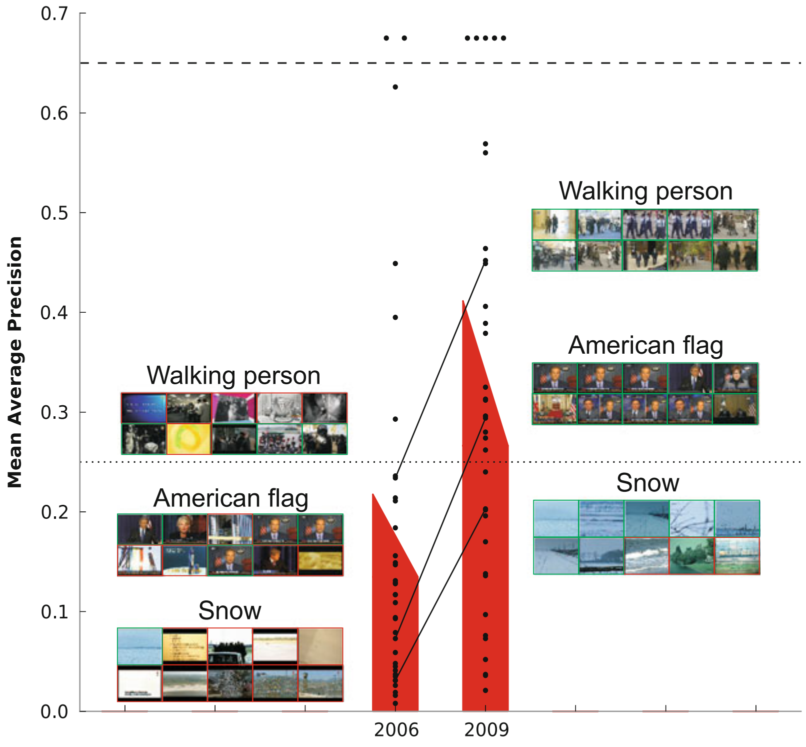
Video retrieval is the process of searching in video based on an analysis of its visual content.

Background

The cause for the general video retrieval problem is the semantic gap: the lack of correspondence between the low-level features that machines extract from the visual signal and the high-level conceptual interpretations a human gives [1]. In order to bridge the gap, many retrieval solutions have been proposed in the past, e.g., by using text, speech, tags, or example images [2, 3]. But the most authentic and cognitive hardest is to type a concept from visual information and to retrieve the images carrying that concept [4].

Theory

The video retrieval method of choice in the field is rendered in Fig. 1. The first step is to extract from an image locally measured features, lots of them, ranging from 40 to 100,000. The features are invariant descriptors which cancel out accidental circumstances of the recording caused by differences in lighting, viewpoint, or scale. In order to capture the complexity of the world, many texture, shape, and color descriptors need to be extracted.



Video Retrieval, Fig. 2 Video retrieval progress as evaluated on 36 concept detectors (●) derived from broadcast video data using state-of-the-art search engines from 2006 and 2009. The figure highlights performance for three typical concepts. The *top* of the skewed bar indicates the maximum average performance by training on similar examples, and the *bottom* indicates the minimum performance when training on a data set of completely different origin [7]. Progress in video retrieval is substantial and quickly maturing in robustness for real-world usage of any concept

The second step is to project the descriptors per pixel onto one of 4,000 words. They are not real words but rather summarizations of one local patch of the image describing one detail: a corner, a texture, or a point. Researchers initially only summarized the image at the most salient points, but it now appears that full-density descriptions are superior.

In the third step, a machine-learning algorithm converts the visual words into one of the semantic concepts. In fact, it assigns a probability to all of the concepts simultaneously, which are used for ranking images in terms of concept presence. Researchers train the algorithm with the help of manually labeled examples. Because there are far more negative examples than positive ones, they intensively compute the optimal machine-learning parameters using grids and GPUs.

Detecting an object such as the American flag is relatively simple if one has an answer to the variance in sensory conditions like illumination and shading, as the flag always shows the same colors and color transitions. Note that a geometrical model of a flag would fail almost always as it rarely appears like a straight square. To detect a walking person from one image requires a richness of poses learned from a labeled set, and snow is even harder to detect as it is white, is texture-free, and may assume all sorts of shapes. Remarkably, although none of the features in current detection methods is specific to any of the concepts, the technique can still detect any of them with sufficient success.

Application

Crucial drivers for progress in video retrieval are international search engine benchmarks such as ImageCLEF (Cross-Language Evaluation Forum), Pascal VOC (Visual Object Classes), PETS (Performance Evaluation of Tracking and Surveillance), and VACE (Video Analysis and Content Extraction). However, thus far the National Institute of Standards and Technology TRECVID (TREC Video Retrieval) benchmark [5] has played the most significant role.

The aim of the TRECVID benchmark is to promote progress in video retrieval by providing a large video collection, uniform evaluation procedures, and a forum for researchers interested in comparing their results. NIST performs an independent examination of results using standard information retrieval evaluation measures, like average precision. With participation from more than 100 teams, including University of Oxford, Tsinghua University, and Columbia University, TRECVID has become the de facto standard for evaluating video retrieval research.

TRECVID has been an important driver for the community in sharing resources for validity of video retrieval experiments, most notably the manual annotations provided by the Large Scale Concept Ontology for Multimedia [6]. Due to the open character of benchmarks, effective concept detection approaches are quickly handed over from one group to another implementing fast convergence on successful methods. It was shown recently that in just 3 years, performance has doubled [7]. For learned concepts, detection rates degenerate when applied to data of a different origin. However, in this setting also, performance has doubled in just 3 years; see Fig. 2.

Open Problems

Despite good progress many problems in video retrieval remain to be solved. Apart from the need to further improve the robustness and efficiency of concept detectors, there is also a need to expand the number of detectors and to facilitate retrieval mechanisms that cater for on-the-fly search [8]. For both scenarios, the widely available socially tagged image examples on the web seem suited. Another open problem considers complex search requests by the combination of detectors at query time into events or short stories. Even though individual detectors may be reasonable, their combination often yields nothing but noise. Finally, the problem of explaining to a user what visual information in the video was the characteristic evidence in making a meaningful retrieval decision is unsolved.

References

1. Smeulders AWM, Worring M, Santini S, Gupta A, Jain R (2000) Content-based image retrieval at the end of the early years. IEEE Trans Pattern Anal Mach Intell 22(12):1349–1380
2. Wactlar HD, Christel MG, Gong Y, Hauptmann AG (1999) Lessons learned from building a terabyte digital video library. IEEE Comput 32(2):66–73
3. Smith JR, Chang SF (1997) Visually searching the web for content. IEEE MultiMed 4(3):12–20
4. Snoek CGM, Worring M (2009) Concept-based video retrieval. Found Trends Inf Retr 4(2):215–322
5. Smeaton AF, Over P, Kraaij W (2006) Evaluation campaigns and TRECVid. In: Proceedings of the ACM SIGMM international workshop on multimedia information retrieval. ACM, New York, pp 321–330
6. Naphade MR, Smith JR, Tešić J, Chang SF, Hsu W, Kennedy LS, Hauptmann AG, Curtis J (2006) Large-scale concept ontology for mul- timedia. IEEE MultiMed 13(3): 86–91
7. Snoek CGM, Smeulders AWM (2010) Visual-concept search solved? IEEE Comput 43(6):76–78
8. Hauptmann AG, Yan R, Lin WH, Christel MG, Wactlar H (2007) Can high-level concepts fill the semantic gap in video retrieval? A case study with broadcast news. IEEE Trans Multimed 9(5):958–966

Video Search

►Video Retrieval

View and Rate-Invariant Human Action Recognition

Vasu Parameswaran[1] and Ashok Veeraraghavan[2]
[1]Microsoft Corporation, Sunnyvale, CA, USA
[2]Department of Electrical and Computer Engineering, Rice University, Houston, TX, USA

Related Concepts

►Gait Recognition; ►Gesture Recognition

Definition

View and rate-invariant human action recognition is the recognition of actions independent of the camera viewpoint, action speed, and frame rate of capture of the video.

Background

At its essence, human action is the movement of the body through a sequence of poses. The visual appearance of a given action in a video sequence depends upon several classes of variables: (1) the geometry of the person performing the action, (2) the style of the action being performed, (3) the clothing worn by the person, (4) the camera viewpoint, and (5) the time taken, not only for the entire action but also for each individual pose transition to complete. A generally applicable human action recognition system needs the ability to classify an action from its visual appearance regardless of the values of the above classes of variables. In other words, the system needs to be *subject invariant*, *style invariant*, *clothing invariant*, *view invariant*, and *rate invariant*. Subject invariance, style invariance and clothing invariance have seen relatively little progress so far. While progress has been made on view invariance and rate invariance, the collective body of work has not yet matured. Therefore, this entry summarizes the main approaches that have been proposed so far, highlighting their motivations, theoretical foundations, and limitations.

Theory

Human action recognition is dependent upon modules isolating different humans in the scene. Therefore, in the following, it is assumed that human detection has been performed and image regions corresponding to a single human performing an action have been identified and are provided as an input to the view and rate-invariant human action recognition system.

View-Invariance

The objective of view invariance arises mainly in the case of monocular video. Multiple cameras allow the possibility of acquiring the depths of image points, which makes the view-invariance problem much easier. Therefore, this section focuses only on monocular video.

Detecting an action requires a model, either explicit or implicit. An explicit model for human action can be based on the temporal sequences of body joint angles. Recovery of joint angles from image sequences is difficult due to joints being physically hidden behind skin and clothing and due to the unknown viewpoint. Although it may be possible to theoretically establish viewpoint invariance using explicit models, such approaches are limited by the difficulties involved in recovering joint angles.

Alternatively, actions can be modeled implicitly, based on quantities derived directly from the observed images. However, such models maintain at best a tenuous link to the hidden joint angles, making it difficult to theoretically establish view invariance.

Given difficulties with both types of approaches, it is not surprising to note that no theoretically sound and practically applicable algorithm has been designed so far that can recognize human actions invariant to viewpoint changes. Nevertheless, two classes of approach have emerged based on explicit and implicit models for human action that are described in more detail below.

Geometry-Inspired Approaches

Geometry-inspired approaches draw upon results from geometric invariance and apply them to points on the body as it moves. Geometry driven approaches depend upon body parts having been detected, tracked, and labeled across frames. Extremities of the body such as the head, hands, and feet have a better chance of being detected, whereas internal joints such as shoulders, elbows, and knees are harder to detect. Therefore, methods based on extremities have a better chance of working in practice than methods based on the full body. However, such methods are limited in applicability due to the exclusion of internal joints. Methods to infer the full 2D body pose from silhouettes have been proposed [2, 11], but a system integrating such methods with a full-body-based view-invariant action recognition method has not been demonstrated yet. An example of an extremity-based approach is [9]. Examples of full-body approaches are [6, 7], and [13].

Image-Based Approaches

An implicit model of an action can be based upon a manifold of appearances arising from the same action as seen from different viewpoints. Given an unknown action from an unknown viewpoint, appearance descriptors are extracted and matched with learned appearance descriptors. Matching error is calculated either based on pixel-wise dissimilarity of the silhouettes [5, 18], or in a derived space such as Hu moment space or Fourier shape space [14].

An implicit model for an action can also be based upon exploiting self-similarity. The observation used in such a method is that although a given pose may appear different from different views, repeating instances of the same pose across time will be self-similar in a given view, provided that the viewing angle between the subject and the camera for the pose is the same. Constructing a pair-wise self-similarity matrix across time will produce structures that appear similar. Such an approach assumes the existence of repeated poses in the action and works best when there are many repeating instances within one action cycle. Examples of such approaches include [3] and [12].

Rate Invariance

A robust action recognition system has to be invariant to the rate at which the action is performed. This will enable the results of the action recognition system to be invariant to the frame rate of the camera and the rate at which the actor is performing these actions. While a linear rate invariance is sufficient to handle frame-rate variations, it is necessary to model and be able to handle nonlinear rate variations in order to handle the rate changes that are due to actors, speed of performance. This is because the actors may perform different subactions at varying relative rates thereby leading to a nonlinear rate change across the entire action.

Motivation

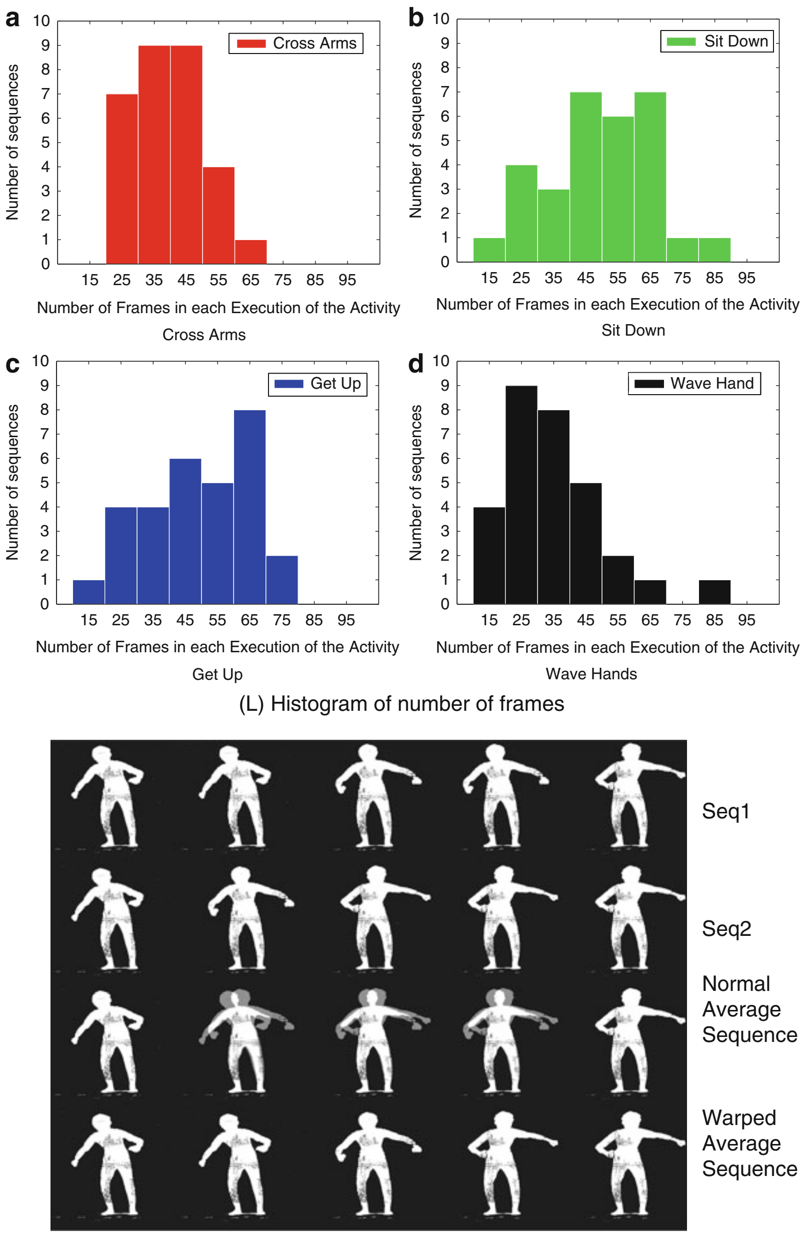
Consider the INRIA iXmas activity recognition dataset [17]. Shown in Fig. 1(L) is the distribution of the number of frames in different executions of the same activity for four distinct activities. Figure 1(L) clearly shows that for the same activity, the rate of execution and consequently the number of frames during the execution varies significantly. Moreover, in most realistic scenarios, this temporal warping might also be inherently nonlinear making simple resampling methods ineffective. This implies that for uncontrolled scenarios, the variations due to temporal warpings could be even more significant. Ignoring this temporal warping might lead to structural inconsistencies apart from providing poor recognition performance. The sequence of images shown in the first two rows of Fig. 1(R) corresponds to two different instances of the same individual performing the same activity. There is an obvious temporal warping between the two sequences. If this temporal warping is ignored, the distance between these two sequences will be large, leading to incorrect matching. Moreover, if some statistical description of the activity, like an average sequence, is required, then ignoring the temporal warping could lead to structural inconsistencies like the presence of four arms and two heads in the average sequence, shown in the third row of Fig. 1(R). If temporal warping is accounted for, then such inconsistencies are avoided and the distance between the two sequences is rightly small. The fourth row shows a typical average sequence obtained after accounting for time warping.

Rate Invariance in Gait Recognition

Results on gait-based person identification shown in [1] indicate that it is very important to take into account the temporal variations in the person's gait. In [15], preliminary work indicating that accounting for execution rate enhances recognition performance for action recognition was presented. Typical approaches for accounting for variations in execution rate are either directly based on the dynamic time-warping (DTW) algorithm [8] or some variation of this algorithm [15]. A method for computing an average shape for a set of dynamic shapes inspite of the existence of varying rates of execution is provided in [4]. A method to learn the best class of time-warping transformations for a given classification problem is proposed in [10].

Decoupling Feature Variability from Rate Variability

The most common way to handle rate variations in the execution of an action is to decouple the variations in features from the variations in dynamics due to rate changes. This is done by modeling an action sequence as a composition of these two sources of variability – variability on the feature space and variability due to execution rate. Keeping the model on the feature space completely independent of the model on the space of execution rates allows for the ability to exploit any of the above-mentioned viewpoint-invariant features. Therefore, as more sophisticated features become available, such models will be able to exploit the characteristics of those features while retaining the ability to deal with variations in execution rate. If the chosen features are viewpoint and anthropometry invariant, then the resulting algorithm becomes invariant to all the three significant modes of variations – viewpoint, anthropometry, and execution rate.



(L) Histogram of number of frames

(R) Two sequences with differing rates of executions and their normal and warped average sequences.

View and Rate-Invariant Human Action Recognition, Fig. 1 Figure courtesy [16] (*L*) Histogram of the number of frames in different executions of the same action in the INRIA iXmas dataset. The histograms for four different activities are shown. (*a*) Cross arms (*b*) Sit down (*c*) Get up (*d*) Wave hands. (*R*) Row 1, Row 2: Two instances of the same activity. Row 3: A simple average sequence. Row 4: average sequence after accounting for time warps

Dynamic Time Warping

Dynamic time warping (DTW) [8] is the most common algorithm that is used for accounting for the nonlinear execution rate variations. The DTW algorithm which is based on dynamic programming computes the best nonlinear time normalization of the test sequence in order to match the template sequence by searching over the space of all time warpings. The advantage of using DTW is that by cleverly using dynamic programming, the complexity of the search space is considerably reduced. The usual temporal consistency constraints used in order to reduce the space of time warpings are:

- End point constraints: The start and the end of the activity trajectories must match exactly.
- Monotonicity: The warping function should be monotonically increasing, i.e., the sequence of action units must be unchanged.

– Continuity: The warping function must be continuous.

The DTW algorithm thus computes the best time warping between a test sequence and a template sequence. Once the best time warping is computed, then the test sequence is then unwarped according to the computed warp. The unwarped action sequences are now time-synchronized since all temporal rate variations have been accounted for. From each frame of the unwarped sequence, features (typically view-invariant features) are extracted and then matched in order to result in an algorithm that is both view and rate invariant.

Open Problems

Approaches that are theoretically view invariant are based on inherent parameters of an action such as joint angles or positions. Unless these methods are coupled with methods that infer the inherent joint parameters, they will not be of practical value. On the other hand, image-based approaches are practical but not generally and probably view invariant. An open problem in view-invariant action recognition is to find a method that is theoretically sound and yet practical.

References

1. Bobick A, Tanawongsuwan R (2003) Performance analysis of time-distance gait parameters under different speeds. In: Proceedings of the 4th international conference on IEEE conference on computer vision and pattern recognition (CVPR), Madison
2. Jiang H (2011) Human pose estimation using consistent max-covering. Pattern Anal Mach Intell IEEE Trans PP(99):1
3. Junejo IN, Dexter E, Laptev I, Perez P (2011) View-independent action recognition from temporal self-similarities. IEEE Trans Pattern Anal Mach Intell 33(1):172–185
4. Maurel P, Sapiro G Dynamic shapes average. www.ima.umn.edu/preprints/may2003/1924.pdf
5. Ogale A, Karapurkar A, Aloimonos Y (2007) View-invariant modeling and recognition of human actions using grammars. In: Vidal R, Heyden A, Ma Yi (eds) Dynamical vision. Volume 4358 of lecture notes in computer science. Springer, Berlin/Heidelberg, pp 115–126
6. Parameswaran V, Chellappa R (2005) Human action recognition using mutual invariants. Comput Vis Image Underst J 98(2):294–324
7. Parameswaran V, Chellappa R (2006) View invariance for human action recognition. Int J Comput Vis 66(1):83–101
8. Rabiner L, Juang B (1993) Fundamentals of speech recognition. Prentice Hall, Englewood Cliffs. http://www.amazon.com/Fundamentals-Speech-Recognition-Lawrence-Rabiner/dp/0130151572
9. Rao C, Yilmaz A, Shah M (2002) View-invariant representation and recognition of actions. Int J Comput Vis 50(2):203–226
10. Ratanamahatana CA, Keogh E (2004) Making time-series classification more accurate using learned constraints. In: Proceedings of the SIAM international conference on data mining, Orlando pp 11–22
11. Rosales R, Sclaroff S (2000) Inferring body pose without tracking body parts. In: Proceedings of the 2000 IEEE conference on computer vision and pattern recognition (CVPR), Hilton Head, Island, vol.2, pp 721–727
12. Seitz SM, Dyer CR (1997) View-invariant analysis of cyclic motion. Int J Comput Vis 25:231–251
13. Shen YP, Foroosh H (2009) View-invariant action recognition from point triplets. IEEE Trans Pattern Anal Mach Intell 31(10):1898–1905
14. Souvenir R, Parrigan K (2009) Viewpoint manifolds for action recognition. EURASIP J Image Video Process
15. Veeraraghavan A, Chellappa R, Roy-Chowdhury AK (2006) The function space of an activity. In: Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR), New York, vol 1, pp 959–968
16. Veeraraghavan A, Srivastava A, Roy-Chowdhury AK, Chellappa R (2009) Rate-invariant recognition of humans and their activities. Image Process IEEE Trans 18(6):1326–1339
17. Weinland D, Ronfard R, Boyer E (2006) Free viewpoint action recognition using motion history volumes. Comput Vis Image Underst 104(2–3):249–257
18. Weinland D, Boyer E, Ronfard R (2007) Action recognition from arbitrary views using 3D exemplars. In: Proceeding of the IEEE 11th international conference on computer vision, ICCV 2007, Rio de Janeiro, 14–21 Oct 2007, pp 1, 7

Vignetting

Amit Agrawal
Mitsubishi Electric Research Laboratories,
Cambridge, MA, USA

Related Concepts

▶Radiance; ▶Radiometric Calibration

Definition

Vignetting is a reduction of an image's brightness at the periphery compared to the center of the image. It describes the effective falloff in irradiance for off-axis points for imaging systems.

Background

In real imaging systems, the image brightness is often reduced at the periphery compared to the center of the image. This effect is known as vignetting and is undesirable for computer vision algorithms that rely on measured pixel intensities. Vignetting can be caused by several mechanisms. The image irradiance varies across the field of view according to the fourth power of the cosine of the field angle. This off-axis illumination falloff is one of the prominent reasons for vignetting and is also referred to as cosine-fourth falloff [1].

Vignetting can also be caused by optical and mechanical effects. Light rays arriving at oblique angles to the optical axis may be obstructed by the aperture stop, lens rim, or improper lens hood. This could lead to spatially varying attenuation over the entire image. A large aperture causes more vignetting, and using a small aperture (increasing the F-number) can reduce it. Real lenses often have multiple lens elements to correct for spherical and chromatic aberrations. Rear lens elements may be partially occluded by front lens elements, reducing off-axis illumination.

For digital cameras, pixel vignetting is another vignetting effect caused by the variation in angular sensitivity of digital sensors to the incoming light. A micro-lens array on top of the sensor helps to collect off-axis light and can reduce pixel vignetting.

Pupil aberration [2] is another cause of irradiance falloff at the periphery. This is attributed to nonlinear refraction of rays resulting in a nonuniform light distribution across the aperture.

Theory

It is important to remove vignetting effects for computer vision applications. A simple way to remove vignetting is to capture a reference photo of a uniformly illuminated white background. This reference photo can be used to cancel the vignetting effect on subsequent images captured by the camera. However, such a reference photo is valid only for the images captured under the same camera settings (zoom, aperture) and the same illumination conditions. This approach cannot be used on photos captured under settings other than those used for the reference photo or on photos captured by unknown cameras. Techniques that model the vignetting effects to recover a vignetting-free image [3–8] are useful in such scenarios (Fig. 1).

Vignetting, Fig. 1 An example of vignetting

Let $\mathbf{x} = [u, v]$ denote a pixel $\mathbf{x}$ with coordinates u and v. The observed image intensities $I(\mathbf{x})$ can be modeled as

$$I(\mathbf{x}) = \mathtt{CRF}(kL(\mathbf{x})V(\mathbf{x})), \qquad (1)$$

where $L(\mathbf{x})$ is the true irradiance, $V(\mathbf{x})$ denotes the attenuation due to vignetting, k is the exposure value, and $\mathtt{CRF}(.)$ is the camera response function. Several models for the vignetting function $V(\mathbf{x})$ exist [3, 5, 6]. The polynomial vignetting model is given by

$$V(\mathbf{x}) = 1 + \sum_{n=1}^{D} \beta_n r(\mathbf{x})^{2n}, \qquad (2)$$

where $r(\mathbf{x})$ is the distance of pixel $\mathbf{x}$ from the image center. Similar model based on hyperbolic cosine functions is described in [7].

In [9], Kang and Weiss proposed a model to correct for vignetting effects using a single photo of a flat, textureless surface. Their model describes the overall image attenuation in terms of an off-axis illumination factor $A(\mathbf{x})$, a camera tilt factor $T(\mathbf{x})$, and a geometric factor $G(\mathbf{x})$:

$$V(\mathbf{x}) = A(\mathbf{x})G(\mathbf{x})T(\mathbf{x}), \qquad (3)$$

where

$$A(\mathbf{x}) = \frac{1}{(1 + r^2(\mathbf{x})/f^2)^2}, \qquad (4)$$

$$G(\mathbf{x}) = 1 - \alpha r(\mathbf{x}), \tag{5}$$

$$T(\mathbf{x}) = \cos\tau(1 + \frac{\tan\tau}{f}(u\sin\psi - v\cos\psi))^3, \tag{6}$$

f denotes the effective focal length and α is a constant describing the geometric factor $G(\mathbf{x})$. The camera tilt factor $T(\mathbf{x})$ is described using a tilt axis in a plane parallel to the image plane at an angle ψ with respect to the x-axis, and the tilt angle is denoted by τ. $T(\mathbf{x})$ is used to account for the foreshortening of the imaged surface with respect to the camera. An extended version of Kang-Weiss model is described in [3]. These models can be used to correct the vignetting effects.

Application

Vignetting is usually an undesirable effect caused by real imaging systems. Due to vignetting, the observed image intensities deviate from those predicted by standard models of image formation. Thus, computer vision algorithms may not perform well in presence of vignetting. On the other hand, photographers sometimes introduce intentional vignetting for artistic effects, such as to draw attention to the center of the frame. Vignetting can also be used for camera calibration under certain conditions [9, 10].

Vignetting models can be used to recover a vignetting-free image. Examples of vignetting removal are presented in [3–5]. Vignetting correction is also used in stitching multiple photos to compensate for the varying intensity of scene points across different images [5, 6].

References

1. Horn B (1986) Robot vision. McGraw-Hill, New York
2. Aggarwal M, Hua H, Ahuja N (2001) On cosine-fourth and vignetting effects in real lenses. In: International conference on computer vision, Vancouver, vol 1, pp 472–479
3. Zheng Y, Lin S, Kang SB (2006) Single image vignetting correction. In: IEEE computer society conference on computer vision and pattern recognition (CVPR), New York, pp 461–468
4. Zheng Y, Yu J, Kang SB, Lin S, Kambhamettu C (2008) Single-image vignetting correction using radial gradient symmetry. In: IEEE computer society conference on computer vision and pattern recognition (CVPR), Anchorage, pp 1–8
5. Kim SJ, Pollefeys M (2008) Robust radiometric calibration and vignetting correction. IEEE Trans Pattern Anal Mach Intell 30:562–576
6. Goldman DB, Chen JH (2005) Vignette and exposure calibration and compensation. In: International conference on computer vision, Beijing, pp 899–906
7. Yu W (2004) Practical anti-vignetting methods for digital cameras. IEEE Trans Consum Electron 50:975–983
8. Yu W, Chung Y, Soh J (2004) Vignetting distortion correction method for high quality digital imaging. In: Proceedings of the 17th IEEE international conference on pattern recognition, Cambridge, pp 666–669
9. Kang SB, Weiss R (2000) Can we calibrate a camera using an image of a flat, textureless lambertian surface? In: European conference on computer vision, Dublin, pp 640–653
10. Zheng Y, Kambhamettu C, Lin S (2009) Single-image optical center estimation from vignetting and tangential gradient symmetry. In: IEEE computer society conference on computer vision and pattern recognition (CVPR), Miami Beach, pp 2058–2065

Vignetting Estimation

►Calibration of Radiometric Falloff (Vignetting)

Viscosity Solution

Fabio Camilli[1] and Emmanuel Prados[2]
[1]Dipartimento SBAI, “Sapienza”, Università di Roma, Rome, Italy
[2]INRIA Rhône-Alpes, Montbonnot, France

Definition

Viscosity solution is a notion of weak solution for a class of partial differential equations of Hamilton-Jacobi type.

Background

A first-order partial differential equation of the type

$$H(x, u(x), Du(x)) = 0 \tag{1}$$

is called a Hamilton-Jacobi equation. A function u is said to be a classical solution of (Eq. 1) over a domain if u is continuous and differentiable over the entire domain and x, $u(x)$, and $Du(x)$ (the gradient of u at x) satisfy the above equation at every point of the domain. Consider the boundary value problem

$$|u'(x)| - 1 = 0 \text{ for } x \in (-1, 1), u(\pm 1) = 0. \quad (2)$$

By Rolle's theorem, it is easily seen that classical solutions of the previous problem do not exist, whereas there exist infinite many weak solutions, that is, continuous functions which satisfy the equation at almost every point (the saw-tooth solutions, see Fig. 1a).

At first, this situation can seem rather atypical, but in fact, it is nothing of the sort. For example, it concerns the distance functions which are widely used in computer vision. In particular, it is easy to see that the distance function of a closed curve in a plan, which typically has strong edges (see Fig. 1), is almost everywhere a solution of the Eikonal equation

$$|Du(x)| - F(x) = 0 \quad (3)$$

with $F(x) = 1$ and $u(x) = 0$ on the curve. Also, in general, as in the case of (Eq. 2), this last equation has no classical solution. Another typical example concerns the shape-from-shading problem which naturally yields a Hamilton-Jacobi equations having the same behavior. In particular, by modeling the problem with an orthographic camera, a directional front lighting, and a Lambertian surface, the problem consists then in solving an Eikonal equation in which the function F depends on the considered image.

It is therefore very important to have a theory which allows merely continuous functions to be solutions of Hamilton-Jacobi equations and to provide at the same time a way to select the relevant solution among the weak solutions of the problem.

Theory

The notion of viscosity solution was introduced at the beginning of the 1980s by M. G. Crandall and P. L. Lions [11], and it is related to Kruzkov's theory of entropy solutions for scalar conservation laws.

The basic idea is to replace the differential $Du(x)$ at a point x where it does not exist (e.g., because of a kink in u) with the differential $D\phi(x)$ of a smooth function ϕ touching the graph of u, from above for the subsolution condition and from below for the supersolution one, at the point x.

Definition 1 *(i) A continuous function u is said to be a viscosity subsolution of* (Eq. 1) *if for any x and for any smooth function ϕ such that $u - \phi$ has a maximum point at x, then*

$$H(x, u(x), D\phi(x)) \leq 0.$$

(ii) A continuous function u is said to be a viscosity supersolution of (Eq. 1) *if for any x and for any smooth function ϕ such that $u - \phi$ has a minimum point at x, then*

$$H(x, u(x), D\phi(x)) \geq 0.$$

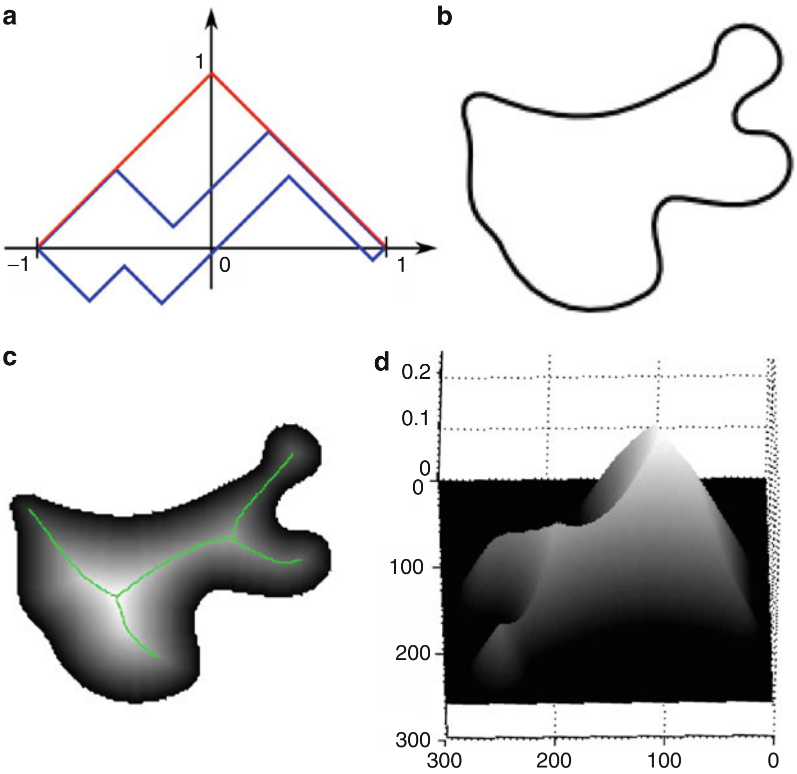
(iii) A continuous function u is said to be a viscosity solution of the Hamilton-Jacobi equation if it is a viscosity subsolution and supersolution.

There is also an equivalent definition for viscosity solution which involves the notion of sub- and superdifferentials (see [5]).

It is straightforward to observe that solutions in the classical sense are viscosity solutions. Inversely, if a viscosity solution u is differentiable at x, then it solves the equation at this point in the classical sense. Hence, the notion of viscosity solution includes the one of classical solution.

By looking closely at the definition, one can understand rather intuitively how it allows a specific weak solution to be selected among the other ones and which of them is selected. In fact, this definition eliminates a certain type of edges. In particular for Hamilton-Jacobi (Eq. 2), the definition allows upward edges, but not downward edges. Thus, in Fig. 1, all the weak solutions of (Eq. 2) which have downward edges are then excluded. Also, only the maximal weak solution (represented in red) is a viscosity solution.

In addition, the definition of viscosity solutions selects among the almost everywhere solutions the one which is consistent with the regularized problem. In fact, the name "viscosity" is motivated by the consistency of the notion with the method of "vanishing viscosity": the viscosity solution of (Eq. 2) can be obtained as the limit for $\epsilon \to 0$ of the classical solutions of



Viscosity Solution, Fig. 1
Distance functions in 1D and 2D: **(a)** three examples of weak solutions of (Eq. 2) The *red curve* corresponds to the viscosity solution. **(b)** Example of a closed curve in the plan. **(c)** Distance function (represented as a color map) to the curve displayed in (**b**). To improve the visibility, the distance function is only displayed inside the curve. As usually, one can distinguish strong edges we have partly highlighted by *green curves*. This gives the skeleton of the shape. **(d)** 3D representation of the function distance (**c**). The distance function is the viscosity solution of the Eikonal Equation (Eq. 3), with $F(x) = 1$ and $u(x) = 0$ on the curve

$$-\epsilon u''(x) + |u'(x)| - 1 = 0 \text{ for } x \in (-1,1),\ u(\pm 1) = 0 \tag{4}$$

(the term ϵ has the physical meaning of a viscosity coefficient).

The main characteristics of the notion of viscosity solution are:

(i) A very efficient and flexible way to prove uniqueness theorems and comparison principles

(ii) General existence results obtained via the adaptation of the classical Perron's method, by approximation arguments (such as the vanishing viscosity method in (Eq. 3)), by means of representation formulas (via dynamic programming methods in optimal control theory), etc.

(iii) The stability of the notion of viscosity solution with respect to the uniform convergence, and its generalizations, which allows to prove, for example, the convergence of numerical schemes

(iv) The correct formulation of various boundary conditions, including the classical Dirichlet, Neumann, and oblique derivative conditions

For good accounts of the viscosity solution theory, we refer to [4, 5, 7, 13]. Finally, let us note that the notion and the theory of viscosity solutions have been extended to a large class of partial differential equations. In particular, a number of results allow us to deal with second-order equations [10], degenerate equations [9], and integro-differential equations [1, 8]. The application to convergence of numerical schemes is also very important; see, for example, [6] and the appendix by Falcone in [5].

Application

The range of applications of the notion of viscosity solution is enormous, including common class of partial differential equations such as evolutive problems and problems with boundary conditions, equations arising in optimal control theory (the Hamilton-Jacobi-Bellman equation), differential games (the Isaacs' equation), second-order equations arising in stochastic optimal control and stochastic differential games, and geometric equations (mean curvature and Monge-Ampere equations).

In computer vision, it has various applications. In particular, the distance functions and the Eikonal equations are widely used. Nowadays, thanks to the links between the viscosity solutions and the optimal control theory [5], one can easily prove that the distance functions correspond to the viscosity solutions of the Eikonal equations, which, moreover, provide various convenient tools for computing them. Also, all these notions have played an important role in shape representation [14, 23], in morphology [2, 21], in tractography [12, 15, 18], and in general, in image processing [3, 18, 22]. Furthermore, they are intensively used in the level set framework where the curves and the surfaces are represented by their signed distance functions [17, 22]. The latter framework is used extensively, for example, in segmentation and 3D reconstruction.

Another main application of the notion of viscosity solution is to shape-from-shading problems, which give rise to first-order differential and integro-differential equations of Hamilton-Jacobi type; see [16, 20], and the entry *Shape from Shading*. A natural question in this context is why the viscosity solutions provide suitable solutions to this specific problem. In other words, why would the viscosity solutions have more sense than any other weak solution? In fact, here, the values of the viscosity solutions mainly come from its amazing stability combined with its consistency with the classical solution. To be more clear, let us consider the shape-from-shading problem with a continuous image I of a real scene. In such a case, to be physically plausible, the real surface u^* behind this image must be smooth (C^1); otherwise, any infinitesimal displacement of the light direction would break this continuity property. Let I^* be the virtual image generated by the considered image formation model with surface u^*. I^* is necessary close to I, if not this would mean that the considered model is not appropriate. Then, thanks to the stability properties, the unique (To be well posed the problem in the viscosity sense, we can assume that we have adequate boundary constraints, for example, only Soner constraints on the boundary of the image if we consider the model of [19].) viscosity solution u to the shape-from-shading Hamilton-Jacobi equation associated with the real image I is close to u^*, because u^* is the viscosity solution to the same equation in which we replace I by I^* (since it is also a solution in the classical sense). In other words, among all the weak solutions of the considered shape-from-shading equation (which has no solution in the classical sense with the real image I because of the modeling errors and the noise), the viscosity solution is necessarily close to the real surface which has been photographed.

References

1. Alvarez O, Tourin A (1996) Viscosity solutions of nonlinear integro-differential equations. Annales de l'institut Henri Poincaré (C) Analyse non linéaire 13(3): 293–317
2. Arehart A, Vincent L, Kimia BB (1993) Mathematical morphology: the Hamilton-Jacobi connection. In: Proceedings of ICCV, Berlin. IEEE Computer Society, pp 215–219
3. Aubert G, Kornprobst P (2006) Mathematical problems in image processing: partial differential equations and the calculus of variations. Applied mathematical sciences. Springer, New York/Secaucus
4. Bardi M, Crandall MG, Evans LC, Soner HM, Souganidis PE (1997) Viscosity solutions and applications. Lecture notes in mathematics, vol 1660. Springer, Berlin. Lectures given at the 2nd C.I.M.E. Session held in Montecatini Terme, June 12–20, 1995. Dolcetta IC, Lions PL (eds). Fondazione C.I.M.E. (C.I.M.E. Foundation)
5. Bardi M, Capuzzo-Dolcetta I (1997) Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations. Systems & control: foundations & applications. Birkhäuser, Boston. With appendices by Maurizio Falcone and Pierpaolo Soravia
6. Barles G, Souganidis PE (1991) Convergence of approximation schemes for fully nonlinear second order equations. Asymptot Anal 4(3):271–283
7. Barles G (1994) Solutions de viscosité des équations de Hamilton-Jacobi. Volume 17 of Mathématiques & Applications (Berlin) (Mathematics & Applications). Springer, Paris
8. Barles G, Imbert C (2008) Second-order elliptic integro-differential equations: viscosity solutions' theory revisited. Annales de l'Institut Henri Poincare (C) Non Linear Analysis 25(3):567–585
9. Camilli F, Siconolfi A (1999) Maximal subsolutions for a class of degenerate Hamilton-Jacobi problems. Indiana Univ Math J 48(3):1111–1132
10. Crandall MG, Ishii H, Lions PL (1992) User's guide to viscosity solutions of second order partial, differential equations. Bull Am Math Soc 27:1–67
11. Crandall MG, Lions P-L (1983) Viscosity solutions of Hamilton-Jacobi equations. Trans Amer Math Soc 277(1):1–42
12. Donnell LO, Haker S, Westin C-F (2002) New approaches to estimation of white matter connectivity in diffusion tensor mri: elliptic pdes and geodesics in a tensor-warped space. In: Dohi T, Kikinis R (eds) Medical image

computing and computer-assisted intervention MICCAI 2002. Lecture notes in computer science, vol 2488. Springer, Berlin/Heidelberg, pp 459–466
13. Fleming WH, Soner HM (2006) Controlled Markov processes and viscosity solutions. Stochastic modelling and applied probability, 2nd edn., vol 25. Springer, New York
14. Kimia BB, Tannenbaum AR, Zucker SW (1994) Shapes, shocks, and deformations i: the components of two-dimensional shape and the reaction-diffusion space. Int J Comput Vis 15:189–224
15. Lenglet C, Prados E, Pons J-P, Deriche R, Faugeras O (2009) Brain connectivity mapping using Riemannian geometry, control theory and PDEs. SIAM J Imaging Sci 2:285–322
16. Lions P-L, Rouy E, Tourin A (1993) Shape-from-shading, viscosity solutions and edges. Numer Math 64(3): 323–353
17. Osher S, Fedkiw R (2002) Level set methods and dynamic implicit surfaces. Applied Mathematics 153, Springer Verlag, New York
18. Pechaud M (2009) Shortest paths calculations, and applications to medical imaging. PhD thesis, University of Paris Diderot
19. Prados E, Faugeras O (2005) Shape from shading: a well-posed problem? In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR'05), San Diego, California, vol II, pp 870–877. IEEE
20. Prados E, Faugeras O (2006) Shape from shading. In: Handbook of mathematical models in computer vision. Springer, New York, pp 375–388
21. Sapiro G, Kimia BB, Kimmel R, Shaked D, Bruckstein AM (1993) Implementing continuous-scale morphology. Pattern Recognit 26(9):1363–1372
22. Sethian JA (1999) Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. Cambridge University Press, Cambridge
23. Tari ZSG, Shah J, Pien H (1997) Extraction of shape skeletons from grayscale images. Comput Vis Image Underst 66:133–146

Visibility Enhancement in Bad Weather

▶Dehazing and Defogging

Vision-Based Control

▶Visual Servoing

Vision-Based Feedback

▶Visual Servoing

Visual Cognition

David Vernon
Informatics Research Centre, University of Skövde,
Skövde, Sweden

Synonyms

Visual Inference

Related Concepts

▶Cognitive Vision

Definition

Visual cognition is the branch of psychology that is concerned with combining visual data with prior knowledge to construct high-level representations and make unconscious decisions about scene content [1].

Background

Although the terms visual cognition and cognitive vision are strikingly similar, they are not equivalent. Cognitive vision refers to goal-oriented computer vision systems that exhibit adaptive and anticipatory behavior. In contrast, visual cognition is concerned with how the human visual system makes inferences about the large-scale composition of a visual scene using partial information [1–3].

Theory

Visual cognition, often associated with high-level vision and top-down visual processing, constructs visual entities by collecting perceived parts into coherent wholes, determining which parts belong together. Since the sensory data on which the processes of visual cognition operate are typically incomplete and insufficient to specify the percept of which we are aware, there are many possible solutions or interpretations. Consequently, additional extraretinal information, often referred to as object information, is used by visual cognition to infer what the percept is.

The entities that are constructed by visual cognition include both static structures, such as perceived surfaces and objects, and dynamic entities that emerge over time, such as patterns of biological motion. The dynamics of these visual entities can be used to infer a causal relationship between events or to attribute some sense of intentionality to the entity. Thus, the unconscious inferences of visual cognition also impact on the construction of a theory of mind for other cognitive agents, i.e., the inference of the goals of other agents [4].

A key role of the representations of visual cognition is their use to communicate with other centers of the brain. Thus, visual cognition provides a bridge to general high-level cognitive function while still making its own independent cognitive inferences. In essence, visual cognition constructs a representation of the visual world which is constantly updated on the basis of new visual data and which encapsulates knowledge about the world in a high-level descriptive manner that can be exchanged with the rest of the brain.

Visual cognition addresses several distinct areas such as visual attention (including spatial attention, selective attention, visual search, change detection, and the control of eye movements) [5–8], short-term and long-term visual memory [9], and object, face, and scene recognition [10, 11].

The processes of visual cognition are held to be principally unconscious, operating rapidly on the flux of visual data sensed by the retina in order to choose the conscious percept of which we become aware. Consequently, visual cognition embraces both the selectivity of visual attention and unconscious inferential decision-making.

Although the primary concern of visual cognition is human visual perception and not computer vision, the two fields share some common ground. For example, many of the theories of visual cognition have their roots in cognitivist psychology which asserts that cognition is intrinsically computational [12, 13]. This has led to several computational models of visual cognition, combining relevant aspects of computer and human vision.

Open Problems

There is some debate in the psychology community as to where one should draw the line between vision and cognition and how sharply one should draw it; for comprehensive discussion, see the paper by Pylyshyn [14], the many commentaries on it (e.g., [15]), and his response [16]. The issue revolves around the cognitive impenetrability of visual perception: whether or not any cognitive functionality such as inference or rationality is involved in visual perception, especially early vision. Cavanagh's recent review [1] suggests that the visual system does have its own independent cognitive processes, quite apart from the more general cognition that occurs in other parts of the brain and with which the processes of visual cognition interact.

References

1. Cavanagh P (2011) Visual cognition. Vis Res 51(13):1538–1551
2. Coltheart V (ed) (2010) Tutorials in visual cognition. Macquarie monographs in cognitive science. Psychology Press, London
3. Pinker S (1984) Visual cognition: an introduction. Cognition 18:1–63
4. Blakemore S, Decety J (2001) From the perception of action to the understanding of intention. Nat Rev Neurosci 2(1):561–567
5. Carrasco M (2011) Visual attention: the past 25 years. Vis Res 51(13):1484–1525
6. Rensink RA (2002) Change detection. Annu Rev Psychol 53:245–277
7. Simons DJ (2000) Current approaches to change blindness. Vis Cogn 7(1–3):1–15
8. Torralba A, Oliva A, Castelhano MS, Henderson JM (2006) Contextual guidance of eye movements and attention in real-world scenes: the role of global features in object search. Psychol Rev 113(4):766–786
9. Deco G, Rolls E (2005) Attention, short term memory, and action selection: a unifying theory. Prog Neurobiol 76: 236–256
10. Spelke ES (1990) Principles of object perception. Cogn Sci 14:29–56
11. Oliva A, Torralba A (2006) Building the gist of a scene: the role of global image features in recognition. Prog Brain Res 155:23–36
12. Newell A (1990) Unified theories of cognition. Harvard University Press, Cambridge
13. Newell A, Simon HA (1976) Computer science as empirical inquiry: symbols and search. Commun Assoc Comput Mach 19:113–126. Tenth turing award lecture, ACM, 1975

14. Pylyshyn ZW (1999) Is vision continuous with cognition? The case for cognitive impenetrability of visual perception. Behav Brain Sci 22(3):341–365
15. Cavanagh P (1999) The cognitive impenetrability of cognition. Behav Brain Sci 22(3):370–371
16. Pylyshyn ZW (1999) Vision and cognition: how do they connect? Behav Brain Sci 22(3):401–414

Visual Cortex Models for Object Recognition

Tomaso Poggio and Shimon Ullman
Department of Brain and Cognitive Sciences,
McGovern Institute, Massachusetts Institute
of Technology, Cambridge, MA, USA

Related Concepts

▸Object Class Recognition (Categorization)

Definition

Visual cortex model-based methods aim to develop algorithms for object detection, representation and recognition that attempt to mimic human visual systems.

Background

Object recognition is difficult Like other natural tasks that our brain performs effortlessly, visual recognition has turned out to be difficult to reproduce in artificial systems. In its general form, it is a highly challenging computational problem which is likely to play a significant role in eventually making intelligent machines. Not surprisingly, it is also an open and key problem for neuroscience.

Within object recognition, it is common to distinguish two main tasks: identification, for instance, recognizing a specific face among other faces, and categorization, for example, recognizing a car among other object classes. We will discuss both of these tasks below, and use "recognition" to include both.

Models of the visual cortex Over the last two decades, some of the best performing recognition systems have come from research at the intersection of computational neuroscience and computer vision. Recent models of visual cortex based directly on known functional anatomy [20, 21] and building on earlier attempts (e.g., [1, 6, 15, 19, 22–24]) were able to account for and predict a number of physiological data from areas of the ventral stream from V1 and V2 to V4 and IT. This family of models was able to mimic human performance in rapid categorization tasks [20]. Surprisingly, some of these models of visual cortex were among the best computer vision systems at the time [16–18, 21].

Computer Vision and the Visual Cortex: Fundamental Differences

The recent past has shown convergence of computational schemes and brain modeling. There still are, however, major differences between models and the cortex, as well as large differences in performance between models and the brain. We will discuss below two examples of prominent features of cortical structure which have only a minor role in current computational models.

Why Hierarchies

The organization of visual cortex is hierarchical, with features of increasing complexity represented at successive layers. Models of the visual cortex have naturally adopted hierarchical structures. In contrast, in computer vision, the large majority of current schemes are nonhierarchical, with no clear difference in performance between hierarchical and nonhierarchical models. Some computational schemes, however, may be implicitly hierarchical and possibly derive some of their power from their hierarchical organization. For instance, SIFT [13] can be regarded as a three-layer network with the output roughly corresponding to intermediate units in a hierarchical cortical model [19]. What is the possible advantage of hierarchical visual representations, and can artificial systems benefit from adopting such representations?

Scale and position invariance One possible role of feature hierarchies is the need to achieve a useful trade-off between selectivity to complex patterns and sufficient tolerance for changes in position and scale, as seen in the response of IT neurons [10–12]. While scale and position invariance can be achieved quite readily in computer vision systems by sequentially scanning the image at different positions and scales, such a strategy appears unlikely to be realized in neural hardware. When properly measured, scale and position tolerance for new objects is less than originally claimed [2], but still substantial [8, 12]: for at least some of the cells in AIT, position tolerance is on the order of 2–4 degrees in the fovea and scale invariance is on the order of a factor of 2–4, which is remarkable large. It is still not entirely clear how such generalization can be achieved without training for different positions and sizes for each object. It appears possible that hierarchical representations make it possible to transfer invariant recognition from previously trained objects to novel objects by the reuse of shared parts at multiple levels.

A second possible advantage of hierarchical representations has to do with efficiency – computational speed and use of computational resources. For instance, hierarchy may increase the efficiency of dealing with multiple classes in parallel, by allowing the use of shared features at multiple levels. An increase in efficiency may also be related to the issue of *sample complexity*. Hierarchical architectures in which each layer is adapted through learning to properties of the visual world may reduce the complexity of the learning task and thus the overall number of labeled examples required for training. Finally, hierarchies also offer an advantage in not only obtaining recognition of the object as a whole but also recognizing and localizing parts and subparts at multiple levels, such as a face together with the eyes, nose, mouth, eyebrow, nostril, upper lip, and the like (see [5]).

Feed Forward vs. Back Projections

A feed-forward architecture from V1 to prefrontal cortex, in the spirit of the Hubel and Wiesel simple-complex hierarchy, seems to account for several properties of visual cells. In particular, recent "readout" experiments measuring information that could be read out from populations of IT cells [8] confirm previous estimates that, after about ~100 ms from onset of the stimulus, performance of the classifier was essentially at its asymptotic performance during passive viewing.

In addition, feed-forward models also appear to account for recognition performance of human subjects for images flashed briefly and followed by a mask [20–22].

The evidence suggests, therefore, that a feed-forward process is sufficient for a fast initial recognition phase, during which primates can already complete difficult recognition tasks involving "what" is in the image. What, then, is the role of the extensive anatomical back projections in the primate visual system?

Their role may be restricted to learning, but we believe it is broader. Even when the feed-forward projections by themselves may be capable of answering the "what" question of vision, the back projections may play a role in answering the detailed question of "what is where" [14]. This proposal is similar to previous ideas suggesting that visual cortex follows a hypothesis-and-verification strategy [3, 7] or a Bayesian inference procedure in which top-down priors are used to compute a set of mutually consistent conditional probabilities at various stages of the visual pathway [9]. A recent model [5] along these lines demonstrated the use of initial classification using a bottom-up sweep, followed by precise localization of the object and its parts and subparts by a top-down pass.

Top-down pathways in the visual cortex also include the dorsal stream and connections between the dorsal and the ventral stream that are likely to be involved in attentional effects. A Bayesian model [4], which takes into account these bottom-up and top-down signals, performs well in recognition tasks and predicts some of the main psychophysical and physiological properties of attention. For natural images, the top-down signal improves object recognition performance and predicts human eye fixations well. The top-down flow of information combined with a hierarchical representation allows the system to answer not only the *what* question, that is, to perform object identification and categorization, but also the *what is where* question, that is, identification and localization at multiple levels. This is a useful addition, but at the same time we do not believe that the process of vision can be fully characterized in terms of answering "what is where" [14]. For example, humans can recognize subtle aspects of actions, goals, and social interactions at a level which is far beyond the capabilities of our present algorithms. They also can answer essentially any reasonable question, beyond what and where, on any given image – in a kind of Turing test for vision. The top-down pathway is likely to play an important role for this broader range of visual tasks.

Discussion: Future

We briefly consider two problem domains for future studies. The first focuses on how to close the gap between computer and human vision in the tasks considered above – object categorization and identification. The second part considers broader aspects of vision and its roots in evolution.

Closing the Performance Gap

One general question regarding possible improvements in visual recognition is whether recognition is obtained by multiple specialized mechanisms or by a uniform scheme applied to different recognition tasks. For example, suggestions have been made that general categorization and individual recognition may be subserved by different mechanisms or that face recognition may depend on special mechanisms, not used for other object categories. It appears to us that the underlying computational problems in different recognition tasks are similar and can therefore be approached by the same general scheme, applied to different training sets (and possibly implemented by more than a single neuronal mechanism). It is also possible that certain cortical regions could specialize in specific categories (such as faces or locations) not because they implement different recognition strategies, but to facilitate selective readout by other cortical regions. The basic recognition scheme could be augmented, however, by specialized mechanisms, dealing with special cases and exceptions. The full system could then be a combination of a scheme that may be characterized as rule based, which can capture the main properties of a category and generalize broadly to novel examples, and a memory-based recognition scheme, which can deal with atypical cases and exceptions to the rule-based scheme.

We next consider future directions which we think could play a useful role in bringing the performance of artificial recognition models closer to the performance level of human vision. These are not the only possible routes for closing the performance gap, but they provide examples of promising general directions motivated by human perception that could usefully be incorporated into artificial systems.

Continuous learning of rich models In current computational schemes, a model for the object or category of interest is constructed during a learning stage and then used for recognition. In contrast, the primate visual system shows continuous plasticity and can continue to learn when confronted with new examples. The disadvantage of a fixed limited training stage is that the resulting object model may remain too simple. A visual category often contains a core of typical examples but also a large number of possible variations, atypical members, and counterexamples. An object can be recognized by its overall shape, but also by small distinguishing parts, and both aspects need to be included in its representation. To achieve human-level performance, it appears therefore that it will be necessary to construct rich object models, learned continuously from a large number of examples.

Such use of continuous learning raises interesting computational challenges: new methods will be required to learn from errors, and to continuously modify an existing representation based on new incoming information, possibly combining rule-based and memory-based mechanisms mentioned above.

Integrating segmentation and recognition Recognition and segmentation are related tasks in the perception of objects: we can usually recognize the object and at the same time identify in the image the precise region containing the object of interest. Historically, segmentation and recognition were treated in computer vision as sequential processes: figure-ground segmentation first identifies in the image a region likely to correspond to a single object; recognition processes are subsequently applied to the selected region to identify the segmented object. More recently, computational models started to treat the two tasks together, performing object segmentation not only in a bottom-up manner based on image properties, but also in a top-down manner based on object representations stored in memory. This led to substantial progress in object segmentation; however, most current recognition systems do not include segmentation as an integral part of the recognition process. It seems to us that recognition and segmentation are closely linked tasks, and their solutions constrain each other. This integration appears to be supported by considerable physiological and psychophysical evidence [25, 26]. A closer integration of recognition and segmentation at both the object and part levels is likely therefore to improve the recognition of objects and their parts.

A Greater Challenge: Vision and Evolution

The brain uses vision, together with other senses, to obtain knowledge about the world and act upon it. This knowledge goes beyond object recognition and categorization: vision is also used, for example, to recognize actions performed by agents in the surrounding environment as well as their goals and social interactions.

It seems to us that these broader aspects of visual recognition cannot be efficiently handled by simple extension of existing recognition methods. It is likely that in addition to the general learning mechanisms currently used in object recognition models, the brain also uses specialized mechanisms, which have evolved to focus on and extract information required for making judgments about actions, goals, social interactions, and the like. Innate structures and circuits in the brain by themselves do not incorporate full solutions to these challenging problems, but are more likely to provide useful constraints and initial biases which later lead, guided by learning from the environment, to powerful specific mechanisms.

A general broad question for future studies is therefore the nature of the innate machinery used by the visual system, its genetic encoding, and how the combination of innate machinery and learning from the environment leads to our understanding of the visual world.

References

1. Amit Y, Mascaro M (2003) An integrated network for invariant visual detection and recognition. Vis Res 43(19): 2073–2088
2. Bruce C, Desimone R, Gross C (1981) Visual properties of neurons in a polysensory area in the superior temporal sulcus of the macaque. J Neurophysiol 46:369–384
3. Carpenter G, Grossberg S (1987) A massively parallel architecture for a self-organizing neural pattern recognition. Comput Vis Graph Image Process 37:54–115
4. Chikkerur S, Serre T, Poggio T (2009) A Bayesian inference theory of attention: neuroscience and algorithms, MIT-CSAIL-TR-2009-047/CBCL-280. Massachusetts Institute of Technology, Cambridge
5. Epshtein B, Lifshitz I, Ullman S (2008) Image interpretation by a single bottom-up top-down cycle. PNAS 105(38):14298–14303
6. Fukushima K (1975) Cognition: a self-organizing multilayered neural network. Biol Cyber 20(3–4):121–136
7. Hawkins J, Blakeslee S (2004) On intelligence. Times Books, New York
8. Hung C, Kreiman G, Poggio T, DiCarlo J (2005) Fast read-out of object identity from macaque inferior temporal cortex. Science 310:863–866
9. Lee TS, Mumford D (2003) Hierarchical Bayesian inference in the visual cortex. J Opt Soc Am A Opt Image Sci Vis 20(7):1434–1448
10. Logothetis NK, Sheinberg DL (1996) Visual object recognition. Ann Rev Neurosci 19:577–621
11. Logothetis NK, Pauls J, Bülthoff HH, Poggio T (1994) View-dependent object recognition by monkeys. Curr Biol 4:401–413
12. Logothetis NK, Pauls J, Poggio T (1995) Shape representation in the inferior temporal cortex of monkeys. Curr Biol 5:552–563
13. Lowe D (2004) Distinctive image features from scale-invariant key-points. Int J Comput Vis 60(2):91–110
14. Marr D (1982) Vision: a computational investigation into the human representation and visual information. W.H. Freeman, New York
15. Mel BW (1997) SEEMORE: combining color, shape and texture histogramming in a neurally-inspired approach to visual object recognition. Neural Comput 9: 777–804
16. Mutch J, Lowe D (2006) Multiclass object recognition using sparse, localized features. In: Proceedings of the IEEE conference on computer vision pattern recognition (CVPR), New York
17. Mutch J, Lowe DG (2008) Object class recognition and localization using sparse features with limited receptive fields. Int J Comput Vis (IJCV) 80(1):45–57
18. Pinto N, Doukhan D, DiCarlo JJ, Cox DD (2009) A high-throughput screening approach to discovering good forms of biologically inspired visual representation. PLoS Comput Biol 5(11):1–12
19. Riesenhuber M, Poggio T (1999) Hierarchical models of object recognition in cortex. Nat Neurosci 2: 1019–1025
20. Serre T, Oliva A, Poggio T (2007) A feedforward architecture accounts for rapid categorization. Proc Natl Acad Sci 104(15):6424
21. Serre T, Kreiman G, Kouh M, Cadieu C, Knoblich U, Poggio T (2007) A quantitative theory of immediate visual recognition. Prog Brain Res 165:33–56
22. Thorpe S (2002) Ultra-rapid scene categorisation with a wave of spikes. In: Second international workshop on biologically motivated computer vision (BMCV), Tübingen, pp 1–15
23. Wallis G, Rolls ET (1997) A model of invariant object recognition in the visual system. Prog Neurobiol 51: 167–194
24. Wersing H, Koerner E (2003) Learning optimized features for hierarchical models of invariant recognition. Neural Comput 15(7):1559–1588

25. Zemel RS, Behrmann M, Mozer MC, Bavelier D (2002) Object recognition processes can and do operate before figure-ground organization. Exp Psychol 28(1):202–217
26. Zhang J, Zisserman A (2006) Dataset issues in object recognition. In: Ponce J, Hebert M, Schmid C, Zisserman A (eds) Toward category-level object recognition. Springer, Berlin, pp 29–48
27. Zhou H, Howard S, Friedman HS, von der Heydt R (2000) Coding of border ownership in monkey visual cortex. J Neurosci 20(17):6594–6611

Visual Hull

David C. Schneider
Image Processing Department, Fraunhofer Heinrich Hertz Institute, Berlin, Germany

Synonyms

Shape from Silhouette

Definition

The visual hull of a three-dimensional object is a bounding volume of the object which is computed from the object's silhouettes in the images of multiple calibrated cameras by intersecting the object's viewing cones.

Background

The concept of visual hull is based on two intuitions: Firstly, the shape of a three-dimensional object is bounded by the silhouette of its projection in an image. Secondly, if several projections from multiple viewpoints are available, these constraints can be combined to obtain an approximation of the object's shape. The practical relevance of visual hull algorithms lies in the fact that they can compute an approximate 3D shape using only silhouettes and calibration data. These are often easier and faster to obtain than image-to-image correspondences, especially for camera configurations with few cameras and/or wide baselines. Visual hull computation is the prototypical example of a shape-from-silhouette method.

The term "visual hull" was coined by Laurentini [1, 2] who studied the properties of hulls obtained from an infinite number of cameras. However, the underlying concepts – the intersection of viewing cones – can be tracked back to the early 1970s (e.g., [3]). Sometimes, the term "visual hull" is restricted to the theoretical case of infinite camera count; hulls computed from a finite number of cameras are then regarded as approximations.

Theory

Definition

The visual hull of an object can be defined either as a surface or as a volume. The following definition is volumetric. Assume that the object can be represented as a surface $\mathcal{S}$ in three-dimensional space. Further assume that there are n cameras viewing the object. The *viewing cone* $\mathcal{C}_i$ of $\mathcal{S}$ with respect to camera $i \in 1 \ldots n$ is a generalized cone defined as the union of all rays that originate from the camera's optical center $\mathbf{c}_i$ and pass through any point $\mathbf{p}$ on $\mathcal{S}$ (Fig. 1, right):

$$\mathcal{C}_i = \bigcup_{\mathbf{p} \in \mathcal{S}, \lambda \geq 0} \lambda \mathbf{p} + (1 - \lambda)\, \mathbf{c}_i$$

The *visual hull* $\mathcal{H}$ of $\mathcal{S}$ is the intersection of all viewing cones (Fig. 1, left):

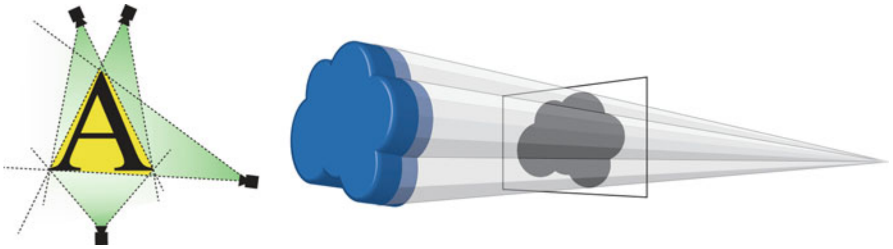
$$\mathcal{H} = \bigcap_{i=1 \ldots n} \mathcal{C}_i$$

Seen as a surface, the visual hull is the boundary of this volume.

Properties

The visual hull bounds the shape it is constructed from, that is, the volumetric intersection of the hull and the original shape is the shape itself. How closely an object's true 3D shape can be approximated by its visual hull depends on two factors:

(1) *The geometry of the object.* The visual hull can represent some but not all concavities of an object. Intuitively, a concavity can be represented if (and as far as) there exists a line through the concavity which does not intersect the object. Or, equivalently, it can be represented if and as far as there exists a location and orientation for a camera such



Visual Hull, Fig. 1 *Left*: The visual hull (*yellow*) of the object (**A**) is the intersection of the four viewing cones (*green*). *Right*: The viewing cone of a 3D object and its relation to the object's silhouette (mask) in an image

that the concavity in the object is represented as a concavity in the silhouette of the object in the camera image. A straightforward example of a representable concavity is the handle of a tea cup. The inside of the cup, on the other hand, is not representable (unless a camera is placed inside the cup).

(2) *The number and placement of the cameras used for computing the hull.* A high count and a regular distribution of cameras are beneficial. With a finite number of cameras, the visual hull typically has a "piecewise extruded" look Fig. 2 shows an example of representable on non-representable geometry in a visual hull reconstruction.

Algorithms

The above definition of the viewing cone only refers to *whether* a ray intersects the surface and not *where*. Therefore, the viewing cones can be computed without knowledge of the shape $\mathcal{S}$ as long as the set of intersecting rays can be computed. Whether a ray of a specific view intersects the surface can be determined from the view's camera matrix and a foreground-background mask, or silhouette, which classifies each pixel as showing a part of $\mathcal{S}$ or the background. Therefore, the visual hull is a shape approximation which can be computed solely from calibrated cameras and a foreground-background segmentation.

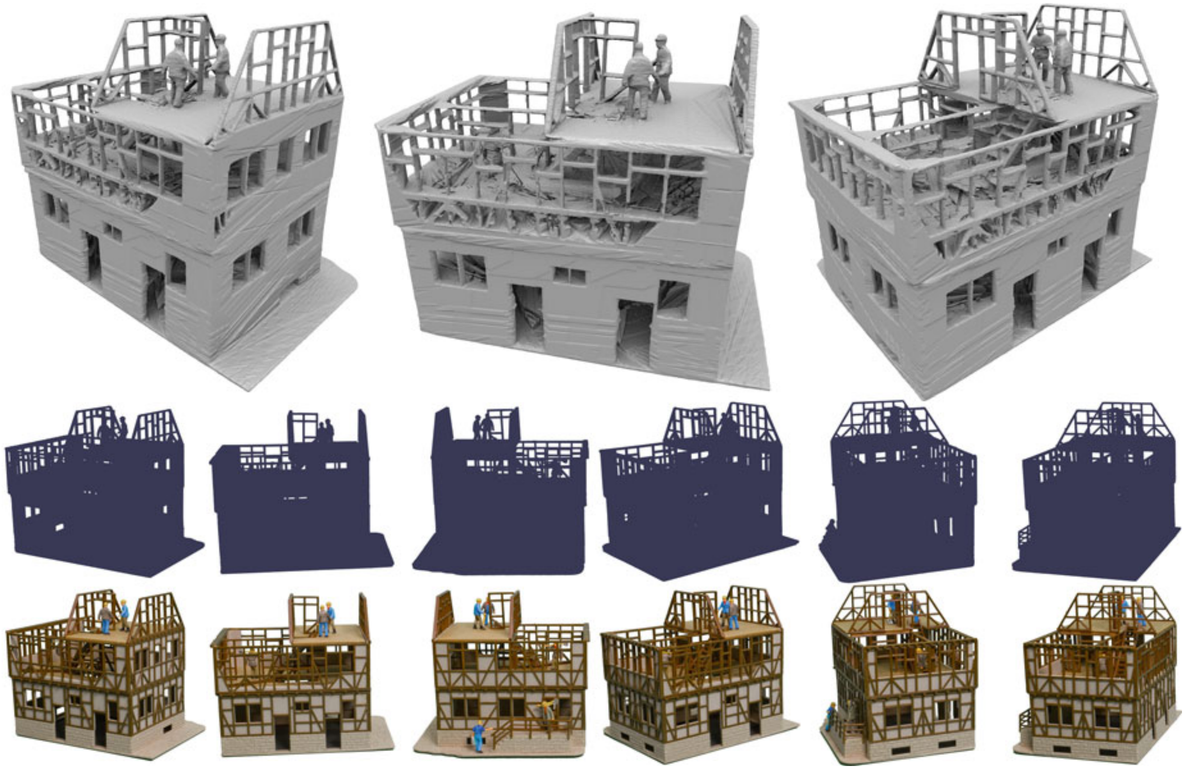
In contrast to the above definition, it is easier to project voxels from the volume into the images than to shoot rays into the volume. The most simple visual hull algorithm is the following: First, all voxels of the volume are marked as "occupied." Then the voxels are traversed and the center of each voxel is projected into all available segmentation masks. A voxel is marked as "unoccupied" if its center projects to a background pixel in *any* of the masks.

For high volume resolutions, traversing all voxels is inefficient with respect to memory usage and computation time. Therefore, many algorithms represent the volume using a hierarchic spatial data structure such as an octree, for example, [4, 5]. When octree cells are projected into the segmentation masks, their spatial extent must be accounted for: An octree cell must be subdivided if any of its projections spans foreground as well as background pixels.

While the volumetric approach makes hull computation relatively simple, it may complicate the further use of the visual hull itself. For many applications, such as rendering on graphics hardware or CAD, a polyhedral surface representation is required. Computing a surface from volume data is a nontrivial task which is often approached with the marching cubes algorithm. The resulting surfaces, however, may require further processing to reduce their high polygon count. This has led to the development of algorithms which directly compute a polyhedral representation of the visual hull, for example, [6, 7].

Application

The visual hull is often used as an initialization and/or as a constraint for 3D reconstruction algorithms. Since the visual hull bounds the true shape, the true shape can be found by removing voxels from the hull. Hull computation, on the other hand, ignores consistency of appearance between multiple views ("photoconsistency"). Therefore, volumetric 3D reconstruction algorithms often aim at minimizing a photoconsistency-based error by removing voxels from the hull. Early examples are space carving [8] or the multi-hypothesis algorithm of [9]. Visual hulls are also used to initialize non-volumetric algorithms such as patch-based stereo [10].



Visual Hull, Fig. 2 Renderings of the visual hull of a toy house (top row). The hull was computed from the masks (middle row) of 216 turntable images of the object (bottom row). Note that fine structures, like the beams, are well represented in the hull as long as they are individuated in the masks. Surface details, on the other hand, cannot be reproduced by the hull

The visual hull is sometimes used directly as shape approximation ("proxy") for image-based rendering and visualization. This approach is particularly successfull for objects where the hull is a good approximation of the actual shape, for example the human body. Hull-based free-viewpoint rendering systems for scenes with humans are described, for example, by [12]. For this type of application, real-time capable hull algorithms have been developed. In [11], for example, new viewpoints are rendered without explicitly representing the geometry of the visual hull.

Obtaining a good segmentation is often a challenge when using visual hull in practice. However, the hull is more sensitive to some types of errors in the masks than others: A voxel is only falsely marked as occupied if the corresponding pixel is falsely classified as foreground in all masks. Conversely, a voxel is falsely marked as unoccupied if a pixel is falsely marked as background in a single mask.

References

1. Laurentini A (1991) The visual hull: a new tool for contour-based image understanding. In: Proceedings of the 7th scandinavian conference on image analysis. Aalborg, Denmark
2. Laurentini A (1994) The visual hull concept for silhouette-based image understanding. IEEE Trans Pattern Anal Mach Intell 16(2):150–162
3. Baumgart BG (1974) Geometric modeling for computer vision. PhD thesis, Stanford, Stanford, CA, USA
4. Ahuja N, Veenstra J (1989) Generating octrees from object silhouettes in orthographic views. IEEE Trans Pattern Anal Mach Intell 11(2):137–149
5. Szeliski R (1993) Rapid octree construction from image sequences. CVGIP 58:23–32
6. Lazebnik S, Furukawa Y, Ponce J (2007) Projective visual hulls. Int J Comput Vis 74:137–165
7. Franco JS, Boyer E (2009) Efficient polyhedral modeling from silhouettes. IEEE Trans Pattern Anal Mach Intell 31(3):414–427
8. Kutulakos KN, Seitz SM (1999) A theory of shape by space carving. In: Proceedings of the seventh IEEE international computer vision conference, Kerkyra, Greece, vol 1, pp 307–314

9. Eisert P, Steinbach E, Girod B (1999) Multi-hypothesis, volumetric reconstruction of 3-d objects from multiple calibrated camera views. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing, Phoenix, AZ, USA, vol 6, pp 3509–3512
10. Furukawa Y, Ponce J (2008) Accurate, dense, and robust multi-view stereopsis. IEEE Trans Pattern Anal Mach Intell 1:1–14
11. Matusik W, Buehler C, Raskar R, Gortler SJ, McMillan L (2000) Image-based visual hulls. In: SIGGRAPH New Orleans, Louisiana, USA
12. Guillemaut J-Y, Hilton A (2011) Joint Multi-Layer Segmentation and Reconstruction for Free-Viewpoint Video Applications, Int J Comput Vis 93(1):73–100

Visual Inference

▸Visual Cognition

Visual Patterns

▸Model-Based Object Recognition

Visual Servoing

François Chaumette
Inria, Rennes, France

Synonyms

Vision-based control; Vision-based feedback

Related Concepts

▸Pan-Tilt-Zoom (PTZ) Camera

Definition

Visual servoing refers to the use of computer vision data as input of real-time closed loop control schemes to control the motion of a dynamic system, a robot typically.

Background

Visual servoing can be seen as sensor-based schema control from a vision sensor. An iterative iteration of the control scheme consists of the 16 following steps: An iteration of the control scheme consists of the following steps:

- Acquire an image.
- Extract some useful image measurements.
- Compute the current value of the visual features used as inputs of the control scheme.
- Compute the error between the current and the desired values of the visual features.
- Update the control outputs, which is usually the robot velocity, to regulate that error to zero, i.e., to minimize its norm.

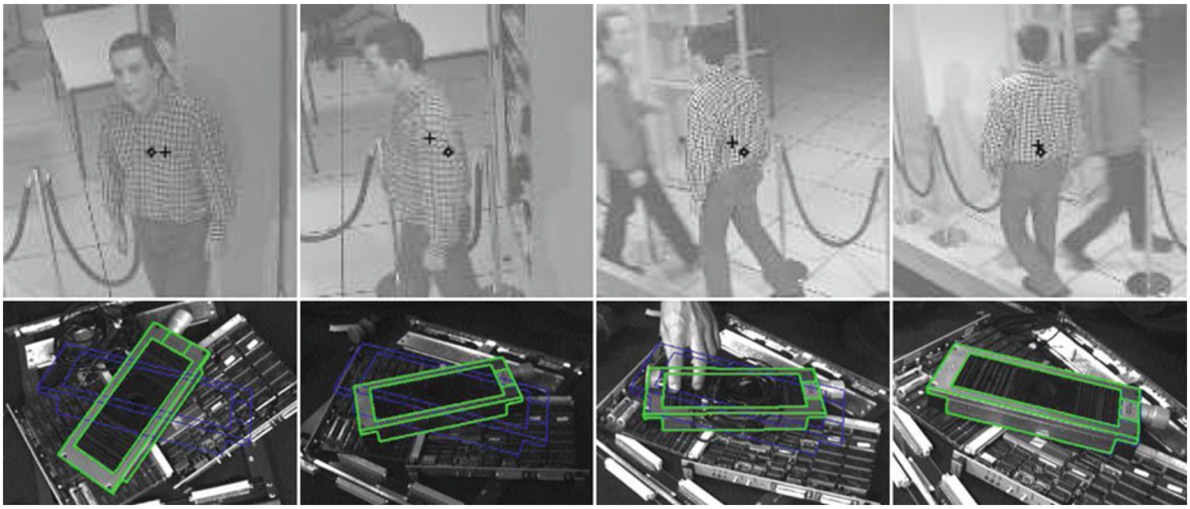
For instance, for the first example depicted on Fig. 1, the image processing part consists in extracting and tracking the center of gravity of the moving people, the visual features are composed of the two Cartesian coordinates of this center of gravity, and the control schemes computes the pan and tilt velocities so that the center of gravity is as near as possible of the image center despite the unknown motion of the people. In the second example where a camera mounted on a six degrees of freedom robot arm is considered, the image measurements are a set of segments that are tracked in the image sequence. From these measurements and the knowledge of the 3D object model, the pose from the camera to the object is estimated and used as visual features. The control scheme now computes the six components of the robot velocity so that this pose reaches a particular desired value corresponding to the object position depicted in blue on the images.

Theory

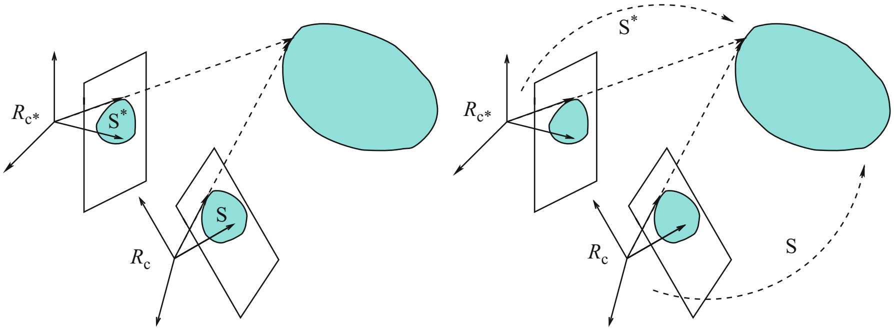
Main if not all visual servoing tasks can be expressed as the regulation to zero of an error $\mathbf{e}(t)$ which is defined by

$$\mathbf{e}(t) = \mathbf{s}(\mathbf{m}(\mathbf{r}(t)), \mathbf{a}) - \mathbf{s}^*(t). \quad (1)$$

The parameters in (Eq. 1) are defined as follows [1]. The vector $\mathbf{m}(\mathbf{r}(t))$ is a set of image measurements (e.g., the image coordinates of interest points, or the



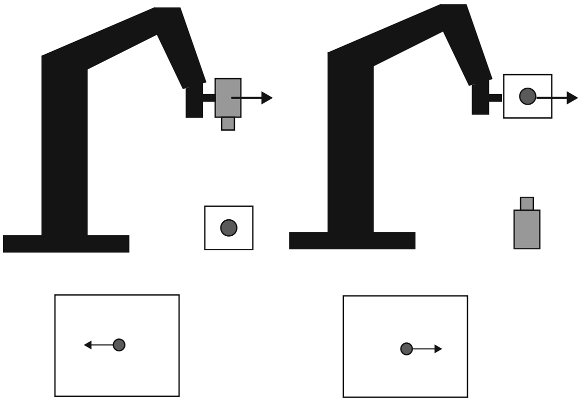
Visual Servoing, Fig. 1 Few images acquired during two visual servoing tasks: on the *top*, pedestrian tracking using a pan-tilt camera; on the *bottom*, controlling the 6 degrees of freedom of an eye-in-hand system so that an object appears at a particular position in the image (shown in *blue*)



Visual Servoing, Fig. 2 If the goal is to move the camera from frame R_c to the desired frame R_{c*}, two main approaches are possible: IBVS on the *left*, where the features **s** and $\mathbf{s}^*$ are expressed in the image, and PBVS on the *right*, where the features **s** and $\mathbf{s}^*$ are related to the pose between the camera and the observed object

area, the center of gravity, and other geometric characteristics of an object). These image measurements depend on the pose $\mathbf{r}(t)$ between the camera and the environment. They are used to compute a vector $\mathbf{s}(\mathbf{m}(\mathbf{r}(t)), \mathbf{a})$ of visual features, in which **a** is a set of parameters that represent potential additional knowledge about the system (e.g., coarse camera intrinsic parameters or 3D model of objects). The vector $\mathbf{s}^*(t)$ contains the desired value of the features, which can be either constant in the case of a fixed goal or varying if the task consists in following a specified trajectory.

Visual servoing schemes mainly differ in the way that the visual features are designed. As represented in Fig. 2, the two most classical approaches are named image-based visual servoing (IBVS), in which **s** consists of a set of 2D parameters that are directly expressed in the image [6, 11], and pose-based visual servoing (PBVS), in which **s** consists of a set of 3D parameters related to the pose between the camera and the target [11, 12]. In that case, the 3D parameters have to be estimated from the image measurements either through a pose estimation process using the knowledge of the 3D target model, or through a partial pose estimation process using the properties of the epipolar geometry between the current and the desired images, or finally through a triangulation process if a stereovision system is considered. Inside IBVS and PBVS approaches, many possibilities exist depending on the choice of the features. Each choice will induce a particular behavior of the system. There also



Visual Servoing, Fig. 3 In visual servoing, the vision sensor can either be mounted on the robot (eye-in-hand configuration) or observing it (eye-to-hand configuration). For the same robot motion, the motion produced in the image will be opposite from one configuration to the other

exist hybrid approaches, named 2-1/2D visual servoing, which combine 2D and 3D parameters in $\mathbf{s}$ in order to benefit from the advantages of IBVS and PBVS while avoiding their respective drawbacks [10].

The design of the control scheme is based on the link between the time variation of the features and the robot control inputs, which are usually the velocity of the robot joints $\mathbf{q}$. This relation is given by

$$\dot{\mathbf{s}} = \mathbf{J_s}\,\dot{\mathbf{q}} + \frac{\partial \mathbf{s}}{\partial t} \tag{2}$$

where $\mathbf{J_s}$ is the features Jacobian matrix, defined from the equation above similarly as the classical robot Jacobian. For an eye-in-hand system (see the left part of Fig. 3), the term $\frac{\partial \mathbf{s}}{\partial t}$ represents the time variation of $\mathbf{s}$ due to a potential object motion, while for an eye-to-hand system (see the right part of Fig. 3), it represents the time variation of $\mathbf{s}$ due to a potential sensor motion.

As for the features Jacobian, in the eye-in-hand configuration, it can be decomposed as [2]

$$\mathbf{J_s} = \mathbf{L_s}\,{}^{c}\mathbf{V}_n\,\mathbf{J}(\mathbf{q}) \tag{3}$$

where

- $\mathbf{J}(\mathbf{q})$ is the robot Jacobian such that $\mathbf{v}_n = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}}$ where $\mathbf{v}_n$ is the robot end effector velocity;
- ${}^{c}\mathbf{V}_n$ is the spatial motion transform matrix from the vision sensor to the end effector. It is given by

$${}^{c}\mathbf{V}_n = \begin{bmatrix} {}^{c}\mathbf{R}_n & [{}^{c}\mathbf{t}_n]_{\times}\,{}^{c}\mathbf{R}_n \\ \mathbf{0} & {}^{c}\mathbf{R}_n \end{bmatrix} \tag{4}$$

  where ${}^{c}\mathbf{R}_n$ and ${}^{c}\mathbf{t}_n$ are, respectively, the rotation matrix and the translation vector between the sensor frame and the end effector frame and where $[{}^{c}\mathbf{t}_n]_{\times}$ is the skew symmetric matrix associated to ${}^{c}\mathbf{t}_n$. Matrix ${}^{c}\mathbf{V}_n$ is constant when the vision sensor is rigidly attached to the end effector, which is usually the case. Thanks to the robustness of closed loop control schemes, a coarse approximation of ${}^{c}\mathbf{R}_n$ and ${}^{c}\mathbf{t}_n$ is sufficient in practice to get an estimation of ${}^{c}\mathbf{V}_n$. If needed, an accurate estimation is possible through classical hand-eye calibration methods.
- $\mathbf{L_s}$ is the interaction matrix of $\mathbf{s}$ defined such that $\mathbf{s} = \mathbf{L_s}\mathbf{v}$ where $\mathbf{v}$ is the relative velocity between the camera and the environment.

In the eye-to-hand configuration, the features Jacobian $\mathbf{J_s}$ is composed of [2]

$$\mathbf{J_s} = -\mathbf{L_s}\,{}^{c}\mathbf{V}_f\,{}^{f}\mathbf{V}_n\,\mathbf{J}(\mathbf{q}) \tag{5}$$

where

- ${}^{f}\mathbf{V}_n$ is the spatial motion transform matrix from the robot reference frame to the end effector frame. It is known from the robot kinematics model.
- ${}^{c}\mathbf{V}_f$ is the spatial motion transform matrix from the camera frame to the reference frame. It is constant as long as the camera does not move. In that case, similarly as for the eye-in-hand configuration, a coarse approximation of ${}^{c}\mathbf{R}_f$ and ${}^{c}\mathbf{t}_f$ is usually sufficient to get an estimation of ${}^{c}\mathbf{V}_f$.

A lot of works have concerned the modeling of the visual features and the determination of the analytical form of the interaction matrix. To give just an example, in the case of an image point with normalized Cartesian coordinates $\mathbf{x} = (x, y)$ and whose 3D corresponding point has depth Z, its interaction matrix is given by [6]

$$\mathbf{L_x} = \begin{bmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{bmatrix} \tag{6}$$

where the three first columns contain the elements related to the three components of the translational velocity, and where the three last columns contain the elements related to the three components of the rotational velocity.

By just changing the parameters representing the same image point, that is, by using the cylindrical coordinates defined by $\rho = \sqrt{x^2+y^2}$ and $\theta = \text{Arctan}(y/x)$, the interaction matrix of these parameters has a completely different form [2]:

$$\mathbf{L}_{(\rho,\theta)} = \begin{bmatrix} -\cos\theta/Z & -\sin\theta/Z & \rho/Z & (1+\rho^2)\sin\theta & -(1+\rho^2)\cos\theta & 0 \\ \sin\theta/(\rho Z) & -\cos\theta/(\rho Z) & 0 & \cos\theta/\rho & \sin\theta/\rho & -1 \end{bmatrix} \quad (7)$$

which implies that using the Cartesian coordinates or the cylindrical coordinates as visual features will induce a different behavior, that is, a different robot trajectory and thus a different trajectory of the image point.

Currently, the analytical form of the interaction matrix is available for most classical geometrical primitives, such as segments, straight lines, ellipses, moments related to planar objects of any shape, and also coordinates of 3D points and pose parameters. Methods also exist to estimate off-line or online a numerical value of the interaction matrix. Omnidirectional vision sensors, the coupling between a camera and structured light, and even 2D echographic probes have also been studied. A large variety of visual features is thus available for many vision sensors.

Once the modeling step has been performed, the design of the control scheme can be quite simple. The most classical control scheme has the following form [2]:

$$\dot{\mathbf{q}} = -\lambda \widehat{\mathbf{J}_s}^{+}(\mathbf{s}-\mathbf{s}^*) + \widehat{\mathbf{J}_s}^{+}\frac{\partial \mathbf{s}^*}{\partial t} - \widehat{\mathbf{J}_s}^{+}\widehat{\frac{\partial \mathbf{s}}{\partial t}} \quad (8)$$

where λ is a positive gain tuning the rate of convergence of the system and $\widehat{\mathbf{J}_s}^{+}$ is the Moore-Penrose pseudoinverse of an approximation or an estimation of the features Jacobian. The exact value of all its elements is indeed generally unknown since it depends of the intrinsic and extrinsic camera parameters, as well as of some 3D parameters such as the depth of the point in (Eqs. 6) and (7).

The second term of the control scheme anticipates for the variation of $\mathbf{s}^*$ in the case of a nonconstant desired value. The third term compensates as much as possible a possible target motion in the eye-in-hand case and a possible camera motion in the eye-to-hand case. They are both null in the case of a fixed desired value and a motionless target or camera. They try to remove the tracking error in the other cases.

Following the Lyapunov theory, the stability of the system can be studied [1]. Generally, visual servoing schemes can be demonstrated to be locally asymptotically stable (i.e., the robot will converge if it starts from a local neighborhood of the desired pose) if the errors introduced in $\widehat{\mathbf{J}_s}$ are not too strong. Some particular visual servoing schemes can be demonstrated to be globally asymptotically stable (i.e., the robot will converge whatever its initial pose) under similar conditions.

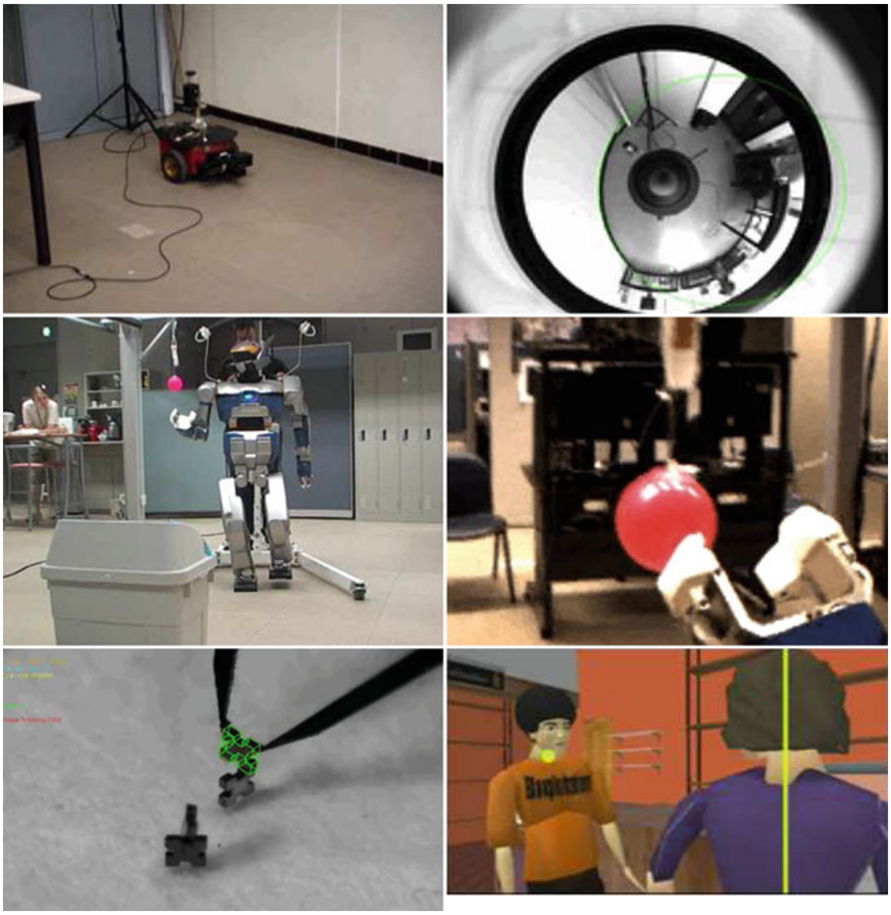
Finally, when the visual features do not constrain all the robot degrees of freedom, it is possible to combine the visual task with supplementary tasks such as joint limits avoidance or the visibility constraint (to be sure that the target considered will always remain in the camera field of view). In that case, the redundancy framework can be applied and the error to be regulated to zero has the following form:

$$\mathbf{e} = \widehat{\mathbf{J}_s}^{+}(\mathbf{s}-\mathbf{s}^*) + (\mathbf{I} - \widehat{\mathbf{J}_s}^{+}\widehat{\mathbf{J}_s})\,\mathbf{e}_2 \quad (9)$$

where $(\mathbf{I} - \widehat{\mathbf{J}_s}^{+}\widehat{\mathbf{J}_s})$ is a projection operator on the null space of the visual task so that the supplementary task $\mathbf{e}_2$ will be achieved at best under the constraint that the visual task is realized [6]. A similar control scheme to (Eq. 8) is now given by

$$\dot{\mathbf{q}} = -\lambda\,\mathbf{e} - \widehat{\frac{\partial \mathbf{e}}{\partial t}} \quad (10)$$

This scheme has for instance been applied for the first example depicted on Fig. 4 where the rotational motion of the mobile robot is controlled by vision while its translational motion is controlled by the odometry to move at a constant velocity.



Visual Servoing, Fig. 4 Few applications of visual servoing: navigation of a mobile robot to follow a wall using an omnidirectional vision sensor (*top line*), grasping a ball with a humanoid robot (*middle line*), assembly of MEMS and film of a dialogue within the constraints of a script in animation (*bottom line*)

Application

Potential applications of visual servoing are numerous. It can be used as soon as a vision sensor is available and a task is assigned to a dynamic system to control its motion. A nonexhaustive list of examples is as follows (see Fig. 4):

- The control of a pan-tilt-zoom camera, as illustrated in Fig. 1 for the pan-tilt case
- Grasping using a robot arm
- Locomotion and dextrous manipulation with a humanoid robot
- Micro- or nanomanipulation of MEMS or biological cells
- Pipe inspection by an underwater autonomous vehicle
- Autonomous navigation of a mobile robot in indoor or outdoor environment
- Aircraft landing
- Autonomous satellite rendezvous
- Biopsy using ultrasound probes or heart motion compensation in medical robotics
- Virtual cinematography in animation

References

1. Chaumette F, Hutchinson S (2006) Visual servo control, part I: basic approaches. IEEE Robot Autom Mag 13(4):82–90
2. Chaumette F, Hutchinson S (2007) Visual servo control, part II: advanced approaches. IEEE Robot Autom Mag 14(1):109–118

3. Chesi G, Hashimoto K (eds) (2010) Visual servoing via advanced numerical methods. LNCIS, vol 401. Springer, Berlin
4. Corke P (1997) Visual control of robots: high-performance visual servoing. Wiley, New York
5. Corke P (2011) Robotics, vision and control. In: Springer tracts in advanced robotics, vol 73. Springer, Berlin
6. Espiau B, Chaumette F, Rives P (1992) A new approach to visual servoing in robotics. IEEE Trans Robot Autom 8(3):313–326
7. Hashimoto K (ed) (1993) Visual servoing: real-time control of robot manipulators based on visual sensory feedback. World Scientific Publishing, Singapore
8. Hutchinson S, Hager G, Corke P (1996) A tutorial on visual servo control. IEEE Trans Robot Autom 12(5): 651–670
9. Kriegman D, Hager G, Morse S (eds) (1998) The confluence of vision and control. LNCIS, vol 237. Springer, London
10. Malis E, Chaumette F, Boudet S (1999) 2-1/2D visual servoing. IEEE Trans Robot Autom 15(2): 238–250
11. Weiss L, Sanderson A, Neuman C (1987) Dynamic sensor-based control of robots with visual feedback. IEEE J Robot Autom 3(5):404–417
12. Wilson W, Hulls C, Bell G (1996) Relative end-effector control using cartesian position-based visual servoing. IEEE Trans Robot Autom 12(5):684–696

Visual SLAM

▸Exploration: Simultaneous Localization and Mapping (SLAM)

Volumetric Texture

▸Bidirectional Texture Function and 3D Texture

von Kries Hypothesis

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

Synonyms

Adaptive gains; Coefficient rule; Color adaptation; Ives transform; von Kries-Ives adaptation

Related Concepts

▸White Balance

Definition

The von Kries hypothesis as applied to chromatic adaptation is an approach that is the basis of most modern color adaptation models. It was first hypothesized by Johannes von Kries in 1902 [3]. The approach requires one to apply a gain to each of the cone responses, independently, so as to keep the adapted appearance of a reference white constant.

Theory

In general terms, let us denote the three cone responses in the human retina as L, M, S, and let the spectral sensitivities of the cones be denoted by $l(\lambda)$, $m(\lambda)$, and $s(\lambda)$. Then for any given stimulus $i_r(\lambda)$ that is incident on the retina, the cone responses are assumed to be given by:

$$\begin{aligned} L &= k \int_\lambda \bar{l}(\lambda) i_r(\lambda) d\lambda \\ M &= k \int_\lambda \bar{m}(\lambda) i_r(\lambda) d\lambda \\ S &= k \int_\lambda \bar{s}(\lambda) i_r(\lambda) d\lambda \end{aligned} \tag{1}$$

with k a normalizing constant, and $\bar{l}(\lambda), \bar{m}(\lambda), \bar{s}(\lambda)$ are the spectral sensitivity functions of the three types of color receptors in the eye. In the above equations, $i_r(\lambda)$ is given by:

$$i_r(\lambda) = i(\lambda) r(\lambda) \tag{2}$$

where $i(\lambda)$ denotes the illuminant and $r(\lambda)$ denotes the spectral reflectivity of a given surface.

The von Kries model is used to predict the response of the cones under changes in the viewing conditions. Specifically, an important change in the viewing conditions is a change in the illuminant. Let us assume that the illuminant $i(\lambda)$ is replaced by a different illuminant, $i_a(\lambda)$. In such a case, let us denote the cone responses by L_a, M_a, and S_a. Then von Kries hypothesis may be expressed as:

$$\begin{bmatrix} L_a \\ M_a \\ S_a \end{bmatrix} = \begin{bmatrix} k_L & 0 & 0 \\ 0 & k_M & 0 \\ 0 & 0 & k_S \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (3)$$

where k_L, k_M, and k_S denote scale factors that are applied independently to the three cone responses. It is perhaps most common to see this above equation represented with $k_L = 1/L_{\max}$, $k_M = 1/M_{\max}$, and $k_S = 1/S_{\max}$ where the max subscript denotes the maximal responses for the LMS cone functions under a normative white stimulus.

In terms of representing the adaptation between the original and new illuminants, in the von Kries model the adapted stimuli are assumed to be given by:

$$\begin{bmatrix} L_a \\ M_a \\ S_a \end{bmatrix} = \begin{bmatrix} \frac{L_{\max a}}{L_{\max}} & 0 & 0 \\ 0 & \frac{M_{\max a}}{M_{\max}} & 0 \\ 0 & 0 & \frac{S_{\max a}}{S_{\max}} \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix}. \quad (4)$$

The von Kries approach, although described in the space of LMS cone functions, can be easily extended to be used in the space of XYZ tristimulus values, defined in terms of colorimetry, rather than in the psychophysically motivated LMS space. And indeed this is typically done, by using a simple 3 × 3 transformation between the LMS cone fundamentals and XYZ color-matching functions:

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \mathbf{M} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \quad (5)$$

The exact values used for matrix **M** depend on the specific cone fundamentals and XYZ color-matching functions chosen [1, 4].

Open Problems

This approach, although not truly accurate with regard to experimental findings in color adaptation, is surprisingly general in its application [2]. Many researchers have been exploring the limitations of this hypothesis and adapting it for use in modern color adaptation models. In Chap. 9 of his book Fairchild provides a comprehensive comparison of the von Kries model to several other color adaptation models developed in the past century [1].

References

1. Fairchild MD (2005) Color appearance models. The Wiley-IS&T series in imaging science and technology, 2nd edn. Wiley, Chichester
2. Hunt RWG (2004) Reproduction of colour, 6th edn. Wiley, Chichester
3. von Kries J (1970) Chromatic adaptation. Festschrift der Albrecht-Ludwig-Universitat, 1902. (trans: MacAdam DL, Sources of color science) MIT, Cambridge
4. Wyszecki G, Stiles WS (1982) Color science: concepts and methods, quantitative data and formulas, 2nd edn. Wiley, New York

von Kries-Ives Adaptation

▶von Kries Hypothesis