A

Action Prototype Trees

▸Prototype-Based Methods for Human Movement Modeling

Action Recognition

▸Activity Recognition
▸Affordances and Action Recognition

Active Calibration

Rui Shen[1], Gaopeng Gou[2], Irene Cheng[1] and Anup Basu[3]
[1]University of Alberta, Edmonton, AB, Canada
[2]Beihang University, Beijing, China
[3]Department of Computing Science, University of Alberta, Edmonton, AB, Canada

Synonyms

Active camera calibration; Pan-tilt camera calibration; Pan-tilt-zoom camera calibration; PTZ camera calibration

Related Concepts

▸Camera Calibration

Definition

Active calibration is a process that determines the geometric parameters of a camera (or cameras) using the camera's controllable movements.

Background

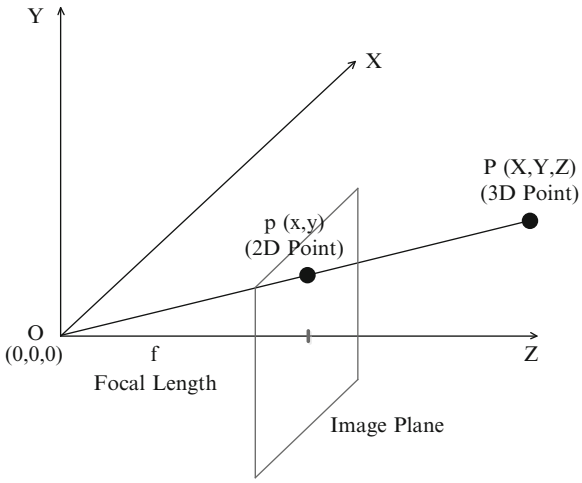
Camera calibration aims to establish the best possible correspondence between the used camera model and the realized image acquisition with a given camera [12], i.e., accurately recover a camera's geometric parameters, such as focal length and image center/principal point, from the captured images. The classical calibration techniques (e.g., [10, 16]) require predefined patterns and static cameras and often involve solving complicated equations. Taking advantage of a camera's controllable movements (e.g., pan, tilt, and roll), active calibration techniques can automatically calibrate the camera.

Theory

The pinhole camera model is one of the most commonly used models, as shown in Fig. 1. $\mathbf{p} = (x, y)^T$ is the 2D projection of the 3D point $\mathbf{P} = (X, Y, Z)^T$ on the image plane. Using homogeneous coordinates, $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{P}}$ have the following relationship:

$$\lambda\tilde{\mathbf{p}} = \mathbf{K}(\mathbf{R} \mid \mathbf{t})\tilde{\mathbf{P}} \quad (1)$$

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



Active Calibration, Fig. 1 The pinhole camera model

Active Calibration, Fig. 2 Canon VC-C50i PTZ camera

where $\mathbf{K} = \begin{pmatrix} f_x & s & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$ is the camera calibration matrix; λ is the depth (when $\mathbf{R}$ is the identity matrix and $\mathbf{t}$ is a zero-vector, $\lambda = Z$); and $\mathbf{R}$ and $\mathbf{t}$ are the 3×3 rotation matrix and the 3×1 translation vector, respectively. f_x and f_y are the focal lengths in pixels in the x- and y-directions, respectively. They are proportional to the focal length f shown in Fig. 1, which is normally measured in millimeters. (x_0, y_0) are the coordinates of the image center/principal point on the image plane, i.e., the intersection of the lens' optical axis and the image plane. s is the skew. Normally, the sensor element is assumed rectangular; therefore, s becomes 0. $\gamma = f_x/f_y$ is the aspect ratio. If the sensor element is quadratic, γ becomes 1. The five parameters in $\mathbf{K}$ are called the camera intrinsic parameters; the three Euler angles in $\mathbf{R}$ and the three offsets in $\mathbf{t}$ are called the camera extrinsic parameters.

Figure 2 shows an active camera. When a camera rotates (no translation), the new image points can be obtained as transformations of the original image points [11]. The camera parameters can be related to the image points before and after rotation and to the angles of rotation. The equations describing these relations are simple and easy to solve. Thus, by considering different movements of the camera (pan, tilt, and roll with measurement of the angles), the intrinsic parameters of a camera can be estimated.

Basu [1] proposed an active calibration technique utilizing the edge information of a static scene. This technique was extended in [2–4] to give more robust estimation of the intrinsic parameters. No special patterns are required, but the observed scene should have strong and stable edges. Four strategies are introduced and validated through experiments [4]. Strategies A and B only utilize pan and tilt movements; strategy C utilizes pan, tilt, and roll movements; strategy D is a special case of strategy C when the roll angle is equal to 180°. The procedure of strategy C is outlined asfollows:

- Using the pan and tilt movement of the camera and a single image contour, obtain the values of f_x and f_y by solving Eqs. 2 and 3.
- Using the roll movement of the camera and a single image contour, obtain the values of δ_x and δ_y by Eqs. 4 and 5.

$$f_y = \frac{\overline{y_t} - \overline{y}(1+\theta_t^2)}{2\theta_t} + \frac{1}{2}\sqrt{\left(\frac{\overline{y}(1+\theta_t^2) - \overline{y_t^2}}{\theta_t}\right)^2 - 4\overline{y^2}} \quad (2)$$

$$f_x = \frac{\overline{x_p} - \overline{x}(1+\theta_p^2)}{2\theta_p} + \frac{1}{2}\sqrt{\left(\frac{\overline{x}(1+\theta_p^2) - \overline{x_p^2}}{\theta_p}\right)^2 - 4\overline{x^2}} \quad (3)$$

where (x_p, y_p) and (x_t, y_t) denote image coordinates after pan and tilt movements, respectively; θ_t and θ_p are the tilt angle and pan angle, respectively.

$$\delta_x(1-\cos(\theta_r)) + \delta_y\left(-\frac{f_x}{f_y}\sin(\theta_r)\right) = \overline{x_r} - \cos(\theta_r)\overline{x} - \frac{f_x}{f_y}\sin(\theta_r)\overline{y} \quad (4)$$

$$\delta_x\left(-\frac{f_x}{f_y}\sin(\theta_r)\right) + \delta_y(1-\cos(\theta_r)) = \overline{y_r} - \cos(\theta_r)\overline{y} + \frac{f_x}{f_y}\sin(\theta_r)\overline{x} \quad (5)$$

where (x_r, y_r) denotes image coordinates after roll movement; (δ_x, δ_y) denotes the error in the estimated principal point (e.g., taking the geometric center of the image plane as the estimated principal point); and θ_r is the roll angle.

Davis and Chen [9] introduced a new pan-tilt camera motion model, in which the pan and tilt axes are not necessarily orthogonal or aligned to the image plane. A tracked object is used to form a large virtual calibration object that covers the whole working volume. A set of pre-calibrated static cameras is needed to record the trajectory. The intrinsic parameters are recovered by minimizing the projection errors between the observed 2D data and the calculated 2D locations of the tracked object using the proposed camera model.

McLauchlan and Murray [13] applied the variable state-dimension filter to calibrating a single camera mounted on a robot by tracking the trajectories of an arbitrary number of tracked corner features and utilizing accurate knowledge of the camera rotation. The camera's intrinsic parameters are updated in real time.

Different zoom settings (focal lengths) can also be employed in active calibration. Seales and Eggert [14] calibrate a camera via a fully automated 4-stage global optimization process using a sequence of images of a known calibration target obtained at different mechanical zoom settings. Collins and Tsin [8] proposed a parametric camera model and calibration procedures for an outdoor active camera system with pan, tilt, and zoom control. Intrinsic parameters are recovered by fitting the camera model with the optic flow produced by the camera's movements. Extrinsic parameters are estimated as a pose estimation problem using sparsely deployed landmarks. Borghese et al. [5] proposed a technique to compute camera focal lengths by zooming a single point, assuming the principal point is in a fixed and known position. Sinha and Pollefeys [15] propose a camera model that incorporates the variation of radial distortion with camera zoom. The intrinsic parameters are first computed at the lowest zoom level from a captured panorama. Then, the intrinsic and radial distortion parameters are estimated at sequentially increased zoom levels, taking into account the influence of camera zoom.

Application

Active calibration has been receiving more and more attention with the increasing use of active systems in various applications, such as object tracking, surveillance, and video conference. More generally, camera calibration is important for any application that involves relating a 2D image to the 3D world. Such applications include pose estimation, 3D motion estimation, automated assembly, close-range photogrammetry, and so on.

Open Problems

Recent research is more focused on automatic active calibration of a multi-camera system without using a predefined calibration pattern/object. Chippendale and Tobia [7] presented an autocalibration system for the estimation of extrinsic parameters of active cameras in indoor environments. One constraint of the camera deployment is that each camera must be able to observe at least one other camera to form an observation chain. The extrinsic parameters are estimated using the circular shape of the camera lenses and a predetermined moving pattern of a particular camera. The accuracy of the algorithm is mainly affected by the distance between cameras. Brückner and Denzler [6] proposed a three-step multi-camera calibration algorithm. The extrinsic parameters of each camera are first roughly estimated using a probability distribution based on the captured images. Then, each camera pair rotates and zooms in a way that maximizes image similarity, and the extrinsic parameters are reestimated based on point correspondence. A final calibration is carried out using the probabilities and the reestimated

Active Calibration, Table 1 Results of image center estimation

Angle (strategy C)	Ground truth		$\sigma = 0$		$\sigma = 5$		$\sigma = 10$	
	δ_x	δ_y	δ_x	δ_y	δ_x	δ_y	δ_x	δ_y
20°	10	20	11	22	12	23	12	23
40°	10	20	11	21	12	22	12	19
60°	10	20	11	21	12	21	13	21
80°	10	20	11	21	11	21	11	22
100°	10	20	10	21	10	21	11	21
120°	10	20	11	21	11	21	11	21
140°	10	20	11	21	11	21	11	21
160°	10	20	10	21	11	21	11	21
180°	10	20	10	20	10	21	10	21
Strategy D	10	20	10	20	11	20	11	20

Active Calibration, Table 2 Results of focal length estimation

Strategy	Ground truth		$\sigma = 0$		$\sigma = 5$	
	f_x	f_y	f_x	f_y	f_x	f_y
Strategy C	400	600	403	602	396	603
Strategy D	400	600	401	601	403	599

extrinsic parameters. This method achieves relatively high accuracy and robustness, but one drawback is the high computational cost.

Experimental Results

Some experimental results using Strategies C and D from [4] are presented below.

Tables 1 and 2 summarize the results of computing the image center and focal lengths using simulated data, along with the ground truths. The simulated data contain an image contour consisting of 50 points. The pan and tilt angles were fixed at 3°. For the experiment on image center calculation, additive Gaussian noise with standard deviation σ of 0 (no noise), 5, and 10 pixels were added to test the robustness of the algorithms. It can be seen that strategy C performs reasonably in determining the image center even when σ is as large as 15. The results of strategy D are similar to those produced by strategy C when the roll angle is 180°.

For the experiment on focal length calculation, additive Gaussian noise with standard deviation σ of 0 (no noise) and 5 pixels were added. Strategy D is a little more accurate as the equations are obtained directly without using the estimates of f_x and f_y. The focal lengths obtained by strategy D are similar to those produced by strategy C. But when the noise is increased, strategy D produces more reliable results than strategy C.

Strategies C and D were also tested on a real camera in an indoor environment. The estimates for δ_x and δ_y obtained by strategy C (90° roll) were 3 and 29 pixels, while the values obtained by strategy D were 2 and 30 pixels, which demonstrates the stability of the active calibration algorithms in real situations. The estimated values of f_x and f_y were 908 and 1,126, respectively.

References

1. Basu A (1993) Active calibration. In: ICRA'93: proceedings of the 1993 IEEE international conference on robotics and automation, Atlanta, vol 2, pp 764–769
2. Basu A (1993) Active calibration: alternative strategy and analysis. In: CVPR'93: proceedings of the 1993 IEEE computer society conference on computer vision and pattern recognition (CVPR), New York, pp 495–500
3. Basu A (1995) Active calibration of cameras: theory and implementation. IEEE Trans Syst Man Cybern 25(2): 256–265
4. Basu A, Ravi K (1997) Active camera calibration using pan, tilt and roll. IEEE Trans Syst Man Cybern B 27(3):559–566
5. Borghese NA, Colombo FM, Alzati A (2006) Computing camera focal length by zooming a single point. Pattern Recognit 39(8):1522–1529
6. Brückner M, Denzler J (2010) Active self-calibration of multi-camera systems. In: Proceedings of the 32nd DAGM conference on pattern recognition, Darmstadt, pp 31–40
7. Chippendale P, Tobia F (2005) Collective calibration of active camera groups. In: AVSS'05: proceedings of the IEEE conference on advanced video and signal based surveillance, Como, pp 456–461
8. Collins RT, Tsin Y (1999) Calibration of an outdoor active camera system. In: CVPR'99: proceedings of the 1999 IEEE computer society conference on computer vision and pattern recognition (CVPR), Ft. Collins, pp 528–534
9. Davis J, Chen X (2003) Calibrating pan-tilt cameras in wide-area surveillance networks. In: ICCV'03: proceedings of the 9th IEEE international conference on computer vision, Nice, pp 144–149
10. Horaud R, Mohr R, Lorecki B (1992) Linear camera calibration. In: ICRA'92: proceedings of the IEEE international conference on robotics and automation, Nice, vol 2, pp 1539–1544
11. Kanatani K (1987) Camera rotation invariance of image characteristics. Comput Vis Graph Image Process 39(3):328–354
12. Klette R, Schlüns K, Koschan A (1998) Computer vision: three-dimensional data from images, 1st edn. Springer, New York/Singapore
13. McLauchlan PF, Murray DW (1996) Active camera calibration for a head-eye platform using the variable state-dimension filter. IEEE Trans Pattern Anal Mach Intell 18(1):15–22

14. Seales WB, Eggert DW (1995) Active-camera calibration using iterative image feature localization. In: CAIP'95: proceedings of the 6th international conference on computer analysis of images and patterns, Prague, pp 723–728
15. Sinha SN, Pollefeys M (2006) Pan-tilt-zoom camera calibration and high-resolution mosaic generation. Comput Vis Image Underst 103(3):170–183
16. Tsai R (1987) A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE J Robot Autom 3(4): 323–344

Active Camera Calibration

▸Active Calibration

Active Contours

▸Numerical Methods in Curve Evolution Theory

Active Sensor (Eye) Movement Control

James J. Clark
Department of Electrical and Computer Engineering, McGill University, Montreal, QC, Canada

Synonyms

Gaze control

Related Concepts

▸Evolution of Robotic Heads; ▸Visual Servoing

Definition

Active sensors are those whose generalized viewpoint (such as sensor aperture, position, and orientation) is under computer control. Control is done so as to improve information gathering and processing.

Background

The *generalized viewpoint* [1] of a sensor is the vector of values of the parameters that are under the control of the observer and which affect the imaging process. Most often, these parameters will be the position and orientation of the image sensor, but may also include such parameters as the focal length, aperture width, and the nodal point to image plane distance, of the camera. The definition of generalized viewpoint can be extended to include illuminant degrees of freedom, such as the illuminant position, wavelength, intensity, spatial distribution (for structured light applications), and angular distribution (e.g., collimation) [2].

Changes in observer viewpoint are used in active vision systems for a number of purposes. Some of the more important uses are:

- Tracking a moving object to keep it in the field of view of the sensor
- Searching for a specific item in the observer's environment
- Inducing scene-dependent optical flow to aid in the extraction of 3D structure of objects and scenes
- Avoiding "accidental" or nongeneric viewpoints, which can result in sensor-saturating specularities or information-hiding occlusions
- Minimizing sensor noise and maximizing novel information content
- Increasing the dynamic range of the sensor, through adjustment of parameters such as sensor sensitivity, aperture, and focus
- Mapping the observer's environment

Theory

LowLevel Camera Motion Control Systems Most robotic active vision control systems act mainly to produce either smooth pursuit motions or rapid saccadic motions. Pursuit motions cause the camera to move so as to smoothly track a moving object, maintaining the image of the target object within a small region (usually in the center) of the image frame. Saccadic motions are rapid, usually large, jumps in the position of the camera, which center the camera field of view on different parts of the scene being imaged. This type of motion is used when scanning a scene,

searching for objects or information, but can also be used to recover from a loss of tracking of an object during visual pursuit.

Much has been learned about the design of pursuit and saccadic motion control systems from the study of primate oculomotor systems. These systems have a rather complicated architecture distributed among many brain areas, the details of which are still subject to vigorous debate [3]. The high-level structure, however, is generally accepted to be that of a feedback system. A very influential model of the human oculomotor control system is that of Robinson [4], and many robotic vision control systems employ aspects of the Robinson model.

The control of an active camera system is both simple and difficult at the same time. Simplicity arises from the relatively unchanging characteristics of the load or "plant" being controlled. For most systems the moment of inertia of the camera changes only minimally over the range of motion, with slight variations arising when zoom lenses are used. The mass of the camera and associated linkages does not change. Inertial effects become more important for control of the "neck" degrees of freedom due to the changing orientation and position of the camera bodies relative to the neck. The specifications on the required velocities and control bandwidth for the neck motions are typically much less stringent than those for the camera motions, so that the inertial effects for the neck are usually neglected. The relatively simple nature of the oculomotor plant means that straightforward proportional-derivative (PD) or proportional-integral-derivative (PID) control systems are often sufficient for implementing tracking or pursuit motion. Some systems have employed more complex optimal control systems (e.g., [5]) which provide improved disturbance rejection and trajectory following accuracy compared to the simpler approaches.

There is a serious difficulty in controlling camera motion systems, however, caused by delays in the control loops. Such delays include the measurement delay due to the time needed to acquire and digitize the camera image and subsequent computations, such as feature extraction and target localization. There is also a delay or latency arising from the time needed to compute the controller output signal [6]. If these delays are not dealt with, a simple PD or PID controller can become unstable, leading to excessive vibration or shaking and loss of target tracking.

There are a number of approaches to dealing with delay. PID or PD systems can be made robust to delays simply by increasing system damping by reducing the proportional feedback gain to a sufficiently low value [7]. This results in a system that responds to changes in target position very slowly, however, and is unacceptable for most applications. For control of saccadic motion, a *sample/hold* can be used, where the position error is sampled at the time a saccade is triggered, and held in a first-order hold (integrator) [8]. In this way, the position error seen by the controller is held constant until the saccadic motion is completed. The controller is insensitive to any changes in the actual target position until the end of this *refractory* period. This stabilizes the controller, but has the drawback that if the target moves during the refractory period, the position error at the end of the refractory period can be large. In this case, another, corrective or secondary, saccadic motion may need to be triggered. For stabilization of pursuit control systems in the presence of delay, an internal positive feedback loop can be employed [4, 8]. This positive feedback compensates for delays in the negative feedback servo loop created by the time taken to acquire an image and compute the target velocity error. The positive feedback loop sends a delayed *efference copy* of the current commanded camera velocity (which is the output of the pursuit controller) back to the velocity error comparator where it is added to the measured velocity error. The positive feedback delay is set so that it arrives at the velocity error comparator at the same time as the measurement of the effect of the current control command, effectively canceling out the negative feedback and producing a new target velocity for the controller. Another delay handling technique is to use *predictive control*, such as the Smith Predictor, where the camera position and controller states are predicted for a time T in the future, where T is the controller delay, and control signals appropriate for those states are computed and applied immediately [6, 7]. Predictive methods make strong assumptions on changes in the external environment (e.g., that all objects in the scene are static or traversing known smooth trajectories). Such methods can perform poorly when these assumptions are violated.

The Next-Look Problem and Motion Planning The control of pursuit and saccadic motions are usually handled by different controllers. While pursuit or tracking behavior can be implemented using frequent small

saccade-like motions, this can produce jumpy images which may degrade subsequent processing operations. With multiple controllers, there needs to be a way for the possibly conflicting commands from the controllers to be integrated and arbitrated. The simplest approach uses the output of the pursuit control system by default, with a switch over to the output of the saccade control system whenever the position error is greater than some threshold and switching back to pursuit control when the position error drops below another (lower) threshold.

Pursuit or tracking of visual targets is just one type of motor activity. Activities such as visual search may require large shifts of camera position to be executed based on a complex analysis of the visual input. The process of determining the active vision system controller set point is often referred to as *sensor planning* [1] or the *next-look problem* [9]. The next-look problem can be interpreted as determining sensor positions which increase or maximize the information content of subsequent measurements. In a visual search task, for example, the next-look may be specified to be a location which is expected to maximally discriminate between target and distractor. One principle that has been successfully employed in next-look processes is that of entropy minimization over viewpoints. In an object recognition or visual search task, this approach takes as the next viewpoint that which is maximally informative relative to the most probable hypotheses [10]. A common approach to the next-view problem in robotic systems is to employ an *attention mechanism* to provide the location of the next view. Based on models of mammalian vision systems, attention mechanisms determine *salient* regions in visual input, which compete or interact in winner-takes-all fashion to select a single location as the target for the subsequent motion [8].

Application

In the late 1980s and early 1990s, commercial camera motion platforms lacked the performance needed by robotics researchers and manufacturers. This led many universities to construct their own platforms and develop control systems for them. These were generally binocular camera systems with pan and tilt degrees of freedom for each camera. Often, to simplify the design, a common tilt action was employed for both cameras, and the pan actions were sometimes linked together to provided vergence and/or version motions only. Examples include the UPenn head [11], generally recognized as the first of its kind, the Harvard head [12], the KTH head [13], the TRISH head from the University of Toronto [14], the Rochester head [15], the SAGEM-GEC-Inria-Oxford head [16], the Surrey head [17], the LIFIA head [18], the LIA/AUC head [19], and the Technion head [5]. These early robotic heads generally used PD servo loops, some with delay compensation mechanisms as described above, and were capable of speeds up to 180 degrees per second. The pan axis maximum rotational velocities were usually higher than those of the tilt and vergence speeds. The axes were most often driven either by DC motors or by stepper motors.

A more recent example of a research system is the head of the iCub humanoid robot [20]. Unlike the early robotic heads, which were one-off systems limited to use in a single laboratory, this robot was developed by a consortium of European institutions and is used in many different research laboratories. It has independent pan and common tilt for two cameras as well as three neck degrees of freedom. The maximum pan speed is 180 degrees per second, and the maximum tilt speed is 160 degrees per second.

Currently, most robotic active vision systems are based on commercially available monocular pan-tilt platforms. The great majority of commercial platforms are designed for surveillance applications and are relatively slow. There are a few systems with specifications that are suitable for robotic active vision systems. Perhaps the most commonly used of these fast platforms are made by FLIR Motion Control Systems, Inc. (formerly Directed Perception). These are capable of speeds up to 120 degrees per second and can handle loads of up to 90 lbs. Commercial systems generally lack torsional motion and hence are not suitable for precise stereo vision applications.

The fastest current commercial pan/tilt units, as well as the early research platforms, only reach maximum speeds of around 200 degrees per second. This is sufficient to match the speeds of human pursuit eye movements, which top out around 100 degrees per second. However, if these speeds are compared to the maximum speed of 800 degrees per second for human saccadic motions, it can be seen that the performance of robotic active vision motion platforms still has room for improvement.

References

1. Tarabanis K, Tsai RY, Allen PK (1991) Automated sensor planning for robotic vision tasks. In: Proceedings of the 1991 IEEE conference on robotics and automation, Sacramento, pp 76–82
2. Yi S, Haralick RM, Shapiro LG (1990) Automatic sensor and light source positioning for machine vision. In: Proceedings of the computer vision and pattern recognition conference (CVPR), Atlantic City, June 1990, pp 55–59
3. Kato R, Grantyn A, Dalezios Y, Moschovakis AK (2006) The local loop of the saccadic system closes downstream of the superior colliculus. Neuroscience 143(1):319–337
4. Robinson DA (1968) The oculomotor control system: a review. Proc IEEE 56(6):1032–1049
5. Rivlin E, Rotstein H (2000) Control of a camera for active vision: foveal vision, smooth tracking and saccade. Int J Comput Vis 39(2):81–96
6. Brown C (1990) Gaze controls with interactions and delays. IEEE Trans Syst Man Cybern 20(1):518–527
7. Sharkey PM, Murray DW (1996) Delays versus performance of visually guided systems. IEE Proc Control Theory Appl 143(5):436–447
8. Clark JJ, Ferrier NJ (1992) Attentive visual servoing. In: Blake A, Yuille AL (eds) An introduction to active vision. MIT, Cambridge, pp 137–154
9. Swain MJ, Stricker MA (1993) Promising directions in active vision. Int J Comput Vis 11(2):109–126
10. Arbel T, Ferrie FP (1999) Viewpoint selection by navigation through entropy maps. In: Proceedings of the seventh IEEE international conference on computer vision, Kerkyra, pp 248–254
11. Krotkov E, Bajcsy R (1988) Active vision for reliable ranging: cooperating, focus, stereo, and vergence. Int J Comput Vis 11(2):187–203
12. Ferrier NJ, Clark JJ (1993) The Harvard binocular head. Int J Pattern Recognit Artif Intell 7(1):9–31
13. Pahlavan K, Eklundh J-O (1993) Heads, eyes and head-eye systems. Int J Pattern Recognit Artif Intell 7(1):33–49
14. Milios E, Jenkin M, Tsotsos J (1993) Design and performance of TRISH, a binocular robot head with torsional eye movements. Int J Pattern Recognit Artif Intell 7(1):51–68
15. Coombs DJ, Brown CM (1993) Real-time binocular smooth pursuit. Int J Comput Vis 11(2):147–164
16. Murray DW, Du F, McLauchlan PF, Reid ID, Sharkey PM, Brady M (1992) Design of stereo heads. In: Blake A, Yuille A (eds) Active vision. MIT, Cambridge, Massachusetts, USA, pp 155–172
17. Pretlove JRG, Parker GA (1993) The Surrey attentive robot vision system. Int J Pattern Recognit Artif Intell 7(1): 89–107
18. Crowley JL, Bobet P, Mesrabi M (1993) Layered control of a binocular camera head. Int J Pattern Recognit Artif Intell 7(1):109–122
19. Christensen HI (1993) A low-cost robot camera head. Int J Pattern Recognit Artif Intell 7(1):69–87
20. Beira R, Lopes M, Praga M, Santos-Victor J, Bernardino A, Metta G, Becchi F, Saltaren R (2006) Design of the robotcub (iCub) head. In: Proceedings of the 2006 IEEE international conference on robotics and automation, Orlando, Florida, USA, pp 94–100

Active Stereo Vision

Andrew Hogue[1] and Michael R. M. Jenkin[2]
[1]Faculty of Business and Information Technology, University of Ontario Institute of Technology, Oshawa, ON, Canada
[2]Department of Computer Science and Engineering, York University, Toronto, ON, Canada

Related Concepts

►Camera Calibration

Definition

Active stereo vision utilizes multiple cameras for 3D reconstruction, gaze control, measurement, tracking, and surveillance. Active stereo vision is to be contrasted with passive or dynamic stereo vision in that passive systems treat stereo imagery as a series of independent static images while active and dynamic systems employ temporal constraints to integrate stereo measurements over time. Active systems utilize feedback from the image streams to manipulate camera parameters, illuminants, or robotic motion controllers in real time.

Background

Stereo vision uses two or more cameras with overlapping fields of view to estimate 3D scene structure from 2D projections. Binocular stereo vision – the most common implementation – uses exactly two cameras, yet one can utilize more than two at the expense of computational speed within the same algorithmic framework.

The "passive" stereo vision problem can be described as a system of at least two cameras attached rigidly to one another with constant intrinsic calibration parameters (assumed), and the stereo pairs are considered to be temporally independent. Thus no assumptions are made, nor propagated, about camera motion within the algorithmic framework. Passive vision systems are limited to the extraction of metric information from a single set of images taken from

different locations in space (or at different times) and treat individual frames in stereo video sequences independently. Dynamic stereo vision systems are characterized by the extraction of metric information from sequences of imagery (i.e., video) and employ temporal constraints or consistency on the sequence (e.g., optical flow constraints). Thus, dynamic stereo systems place assumptions on the camera motion such as its smoothness (and small motion) between subsequent frames. Active stereo vision systems subsume both passive and dynamic stereo vision systems and are characterized by the use of robotic camera systems (e.g., stereo heads) or specially designed illuminant systems (e.g., structured light) coupled with a feedback system (see Fig. 1) for motor control. Although systems can be designed with more modest goals – object tracking, for example – the common computational goal is the construction of large-scale 3D models of extended environments.

Theory

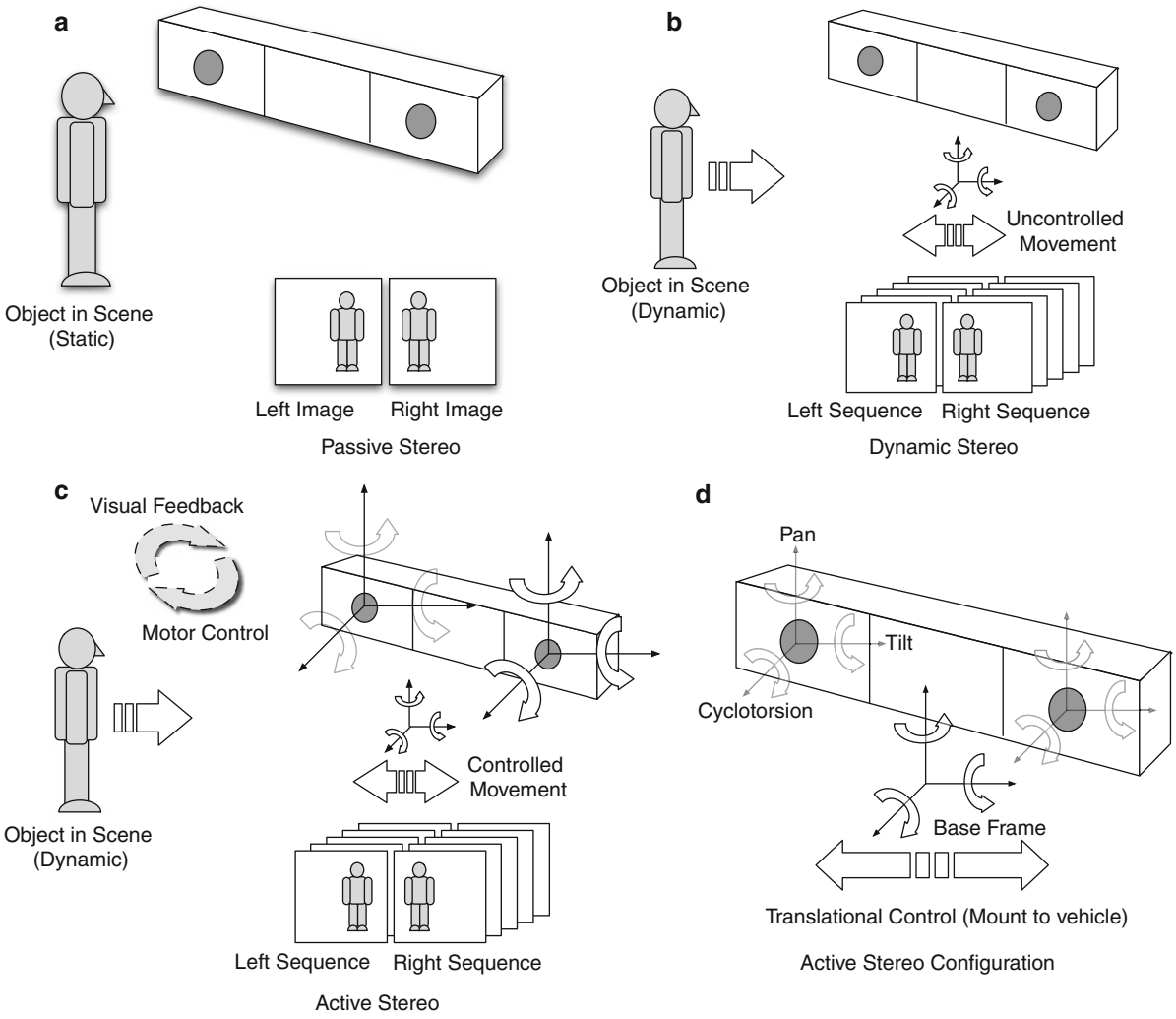
Fundamentally, active stereo systems (see [1]) must solve three rather complex problems: (1) spatial correspondence, (2) temporal correspondence, and (3) motor/camera/illuminant control. Spatial correspondence is required in order to infer 3D depth information from the information available in camera images captured at one time instant, while temporal correspondence is necessary to integrate visual information over time. The spatial and temporal correspondences can either be treated as problems in isolation or integrated within a common framework. For example, stereo correspondence estimation can be seeded using an ongoing 3D representation using temporal coherence (e.g., [2, 3]) or considered in isolation using standard disparity estimation algorithms (see [4]).

Motor or camera control systems are necessary to move (rotate and translate) the cameras so they look in the appropriate direction (i.e., within a tracking or surveillance application), change their intrinsic camera parameters (e.g., focal length or zoom), or to tune the image processing algorithm to achieving higher accuracy for a specific purpose. Solving these three problems in an active stereo system enables one to develop "intelligent" algorithms that infer egomotion [5], autonomously control vehicles throughout the world [6], and/or reconstruct 3D models of the environment [7, 8]. Examples of the output of such a system is shown in Fig. 2, and [9] provides an example of an active system that interleaves the vergence and focus control of the cameras with surface estimation. The system uses an adaptive self-calibration method that integrates the estimation of camera parameters with surface estimation using prior knowledge of the calibration, the motor control, and the previous estimate of the 3D surface properties. The resulting system is able to automatically fixate on salient visual targets in order to extend the surface estimation volume.

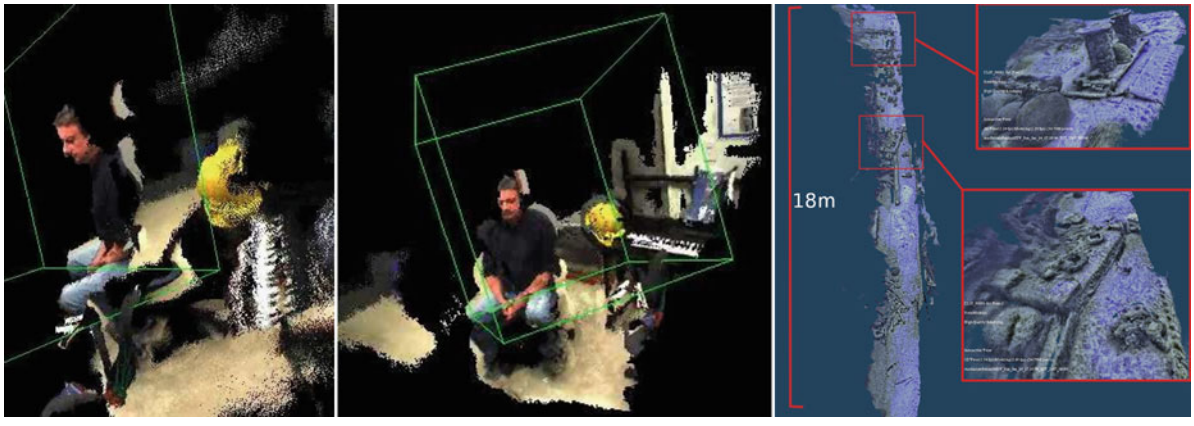
Although vision is a powerful sensing modality, it can fail. This is a critical issue for active stereo vision where data is integrated over time. The use of complementary sensors – traditionally Inertial Measurement Units (see [10]) – augment the camera hardware system with the capability to estimate the system dynamics using real-world constraints. Accelerometers, gyroscopes, and compasses can provide timely and accurate information either to assist in temporal correspondences and ego-motion estimation or as a replacement when visual information is unreliable or absent (i.e., dead reckoning).

Relation to Robotics and Mapping

A wide range of different active and dynamic stereo systems have been built (e.g., [7, 8, 11, 12]). Active systems are often built on top of mobile systems (e.g., [7]) blurring the distinction between active and dynamic systems. In robotics, active stereo vision has been used for vehicle control in order to create 2D or 3D maps of the environment. Commonly the vision system is complemented by other sensors. For instance in [13], active stereo vision is combined with sonar sensors to create 2D and 3D models of the environment. Murray and Little [14] use a trinocular stereo system to create occupancy maps of the environment for in-the-loop path planning and robot navigation. Diebel et al. [15] employ active stereo vision for simultaneous estimation of the robot location and 3D map construction, and [7], describes a vision system used for in-the-loop mapping and navigational control for an aquatic robot. Davison, in [6], was one of the first to effectively demonstrate the use of active stereo vision technology as part of the navigation loop. The system used a stereo head to selectively fixate scene features that improve the quality of the estimated map and trajectory. This involved using knowledge of the



Active Stereo Vision, Fig. 1 Different types of stereo systems



Active Stereo Vision, Fig. 2 Point cloud datasets obtained by the active stereo system described in [7]

current map of the environment to point the camera system in the direction where it should find salient features that it had seen before, move the robot to a location where the features are visible, and then searching visually to find image locations corresponding to these features.

Active Stereo Heads

The development of hardware platforms to mimic human biology has resulted in a variety of different designs and methods for controlling binocular sets of cameras. These result in what is known as "stereo heads" (see entry on *the evolution of stereo heads* in this volume). These hardware platforms all have a common set of constraints, i.e., the systems consist of two cameras (binocular) with camera intrinsics/extrinsics that may be controlled. In [16], an active stereo vision system is developed that mimics human biology that uses a bottom-up saliency map model coupled with a selective attention function to select salient image regions. If both left and right cameras estimate similar amounts of saliency in the same areas, the vergence of the cameras are controlled so that the cameras are focused on this particular landmark.

Autocalibration

A fundamental issue with active stereo vision is the need to establish and maintain calibration parameters online. Intrinsics and extrinsics are necessary to the 3D estimation process as they define the epipolar constraints which enable efficient disparity estimation algorithms [17, 18]. Each time the camera parameters are modified (e.g., vergence of the cameras, change of focus), the epipolar geometry must be re-estimated. Although kinematic modeling of motor systems provide good initial estimates of changes in camera pose, this is generally insufficiently accurate to be used by itself to update camera calibration. Thus, autocalibration becomes an important task within active stereo vision. Approaches to autocalibration are outlined in [17, 19]. In [17], the autocalibration algorithm operates on pairs of stereo images taken in sequence. A projective reconstruction for motion and structure of the scene is constructed. This is performed for each pair of stereo images individually for the same set of features (thus they must be matched in the stereo pairs as well as tracked temporally). The projective solutions can be upgraded to an affine solution (ambiguous up to a rigid rotation/translation/scale) by noting these features should match in 3D space as well as in 2D space. A transformation can be linearly estimated that constrains the projective solution to an affine reconstruction. Once the plane at infinity is known, the affine solution may be upgraded to a metric solution. In order to achieve the desired accuracy in the intrinsics, a nonlinear minimization scheme is employed to improve the solution. If one trusts the accuracy of the camera motion control system, the extrinsics can be seeded with this information in a nonlinear optimization scheme that minimizes the reprojection error of the image matching points and their 3D triangulated counterparts. This nonlinear optimization is known as bundle adjustment [20] and is used in a variety of forms in the structure-from-motion literature (see [17, 18]).

Relation to Other Types of Stereo Systems

Since active stereo systems are characterized by the use of visual feedback to inform motor control systems (or higher-level vehicular navigational systems), they are related to a wide range of research areas and hardware systems. Mounting a stereo system to a robotic vehicle is common in the robotics literature to inform the navigation system about the presence of obstacles [21] and to provide input to mapping algorithms [22]. The use of such active systems are applicable directly to autonomous systems as they provide a high amount of controllable accuracy and dense measurements at relatively low computational cost. One significant example is the use of active stereo in the Mars Rover autonomous vehicles [12].

Estimating 3D information from stereo views is problematic due to the lack of (or ambiguous) texture in man-made environments. This can be alleviated with the use of active illumination [23]. Projecting a known pattern, rather than uniform lighting, into the scene enables the estimation of a more dense disparity field using standard stereo disparity estimation algorithms due to the added texture in textureless regions (see [24]). The illumination may be controlled actively depending on perceived scene texture, the desired range, or the ambient light intensity of the environment. The illumination may be within the visible light spectrum or in the infrared spectrum as most camera sensors are sensitive to IR light. This has the added advantage that humans in the environment are not affected by the additional illumination.

Application

Active stereo vision is characterized by the use of visual feedback in multi-camera systems to control the intrinsics and extrinsics of the cameras (or vehicular platforms). Active stereo vision systems find a wide range of application in autonomous vehicle navigation, gaze tracking, and surveillance. A host of hardware systems exist and commonly utilize two cameras for binocular stereo and motors to control the gaze/orientation of the system. Visual attentive processes (e.g., [25, 26]) may be used to determine the next viewpoint for a particular task, and dense stereo algorithms can be used for estimating 3D structure of the scene. Fundamental computational issues include autocalibration of the sensor with changes in its configuration and the development of active stereo control and reconstruction algorithms.

References

1. Vieville T (1997) A few steps towards 3d active vision. Springer, New York/Secaucus
2. Leung C, Appleton B, Lovell B, Sun C (2004) An energy minimisation approach to stereo-temporal dense reconstruction. In: Proceedings of the 17th international conference on pattern recognition, vol 4, Cambridge, pp 72–75
3. Min D, Yea S, Vetro A (2010) Temporally consistent stereo matching using coherence function. In: 3DTV-conference: the true vision – capture, transmission and display of 3D video (3DTV-CON), 2010, Tampere, pp 1–4
4. Scharstein D, Szeliski R (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int J Comput Vis 47(1–3):7–42
5. Olson CF, Matthies LH, Schoppers M, Maimone MW (2001) Stereo ego-motion improvements for robust rover navigation. In: Proceedings IEEE international conference on robotics and automation, vol 2, Seoul, pp 1099–1104
6. Davison AJ (1998) Mobile robot navigation using active vision. PhD thesis, University of Oxford
7. Hogue A, Jenkin M (2006) Development of an underwater vision sensor for 3d reef mapping. In: IEEE/RSJ international conference on intelligent robots and systems, Beijing, pp 5351–5356
8. Se S, Jasiobedzki P (2005) Instant scene modeler for crime scene reconstruction. In: 2005 IEEE computer society conference on computer vision and pattern recognition workshop on safety and security applications, vol III, San Diego, pp 123–123
9. Ahuja N, Abbott A (1993) Active stereo: integrating disparity, vergence, focus, aperture and calibration for surface estimation. IEEE Trans Pattern Anal Mach Intell 15(10): 1007–1029
10. Everett HR (1995) (ECCV) Sensors for mobile robots: theory and application. A. K. Peters, Ltd., Natick
11. Grosso E, Tistarelli M, Sandini G (1992) Active/dynamic stereo for navigation. In: Second European conference on computer vision, Santa Margherita Ligure, pp 516–525
12. Maimone MW, Leger PC, Biesiadecki JJ (2007) Overview of the mars exploration rovers autonomous mobility and vision capabilities. In: IEEE international conference on robotics and autonomsou (ICRA) space robotics workshop, Rome
13. Wallner F, Dillman R (1995) Real-time map refinement by use of sonar and active stereo-vision. Robot Auton Syst 16(1):47–56. Intelligent robotics systems SIRS'94
14. Murray D, Little JJ (2000) Using real-time stereo vision for mobile robot navigation. Auton Robot 8(2): 161–171
15. Diebel J, Reutersward K, Thrun S, Davis J, Gupta R (2004) Simultaneous localization and mapping with active stereo vision. In: IEEE/RSJ international conference on intelligent robots and systems, vol 4, Sendai, pp 3436–3443
16. Jung BS, Choi SB, Ban SW, Lee M (2004) A biologically inspired active stereo vision system using a bottom-up saliency map model. In: Rutkowski L, Siekmann J, Tadeusiewicz R, Zadeh LA (eds) Artificial intelligence and soft computing – ICAISC 2004. Volume 3070 of lecture notes in computer science. Springer, Berlin/Heidelberg, pp 730–735
17. Hartley RI, Zisserman A (2000) Multiple view geometry in computer vision. Cambridge University Press, Cambridge
18. Faugeras O, Luong QT, Papadopoulou T (2001) The geometry of multiple images: the laws that govern the formation of images of a scene and some of their applications. MIT, Cambridge
19. Horaud R, Csurka G (1998) Self-calibration and euclidean reconstruction using motions of a stereo rig. In: Sixth international conference on computer vision, Bombay, pp 96–103
20. Triggs B, McLauchlan P, Hartley R, Fitzgibbon A (2000) Bundle adjustment – a modern synthesis. In: Triggs B, Zisserman A, Szeliski R (eds) Vision algorithms: theory and practice. Volume 1883 of lecture notes in computer science. Springer, New York, pp 298–372
21. Williamson T, Thorpe C (1999) A trinocular stereo system for highway obstacle detection. In: IEEE international conference on robotics and automation, Detroit, pp 2267–2273
22. Schleicher D, Bergasa LM, Ocaña M, Barea R, López E (2010) Real-time hierarchical stereo visual slam in large-scale environments. Robot Auton Syst 58:991–1002
23. Se S, Jasiobedzki P, Wildes R (2007) Stereo-vision based 3d modeling of space structures. In: Proceedings of the SPIE conference on sensors and systems for space applications, vol 6555, Orlando
24. Rusinkiewicz S, Hall-Holt O, Levoy M (2002) Real-time 3d model acquisition. ACM Trans Graph 21(3):438–446
25. Frintrop S, Rome E, Christensen HI (2010) Computational visual attention systems and their cognitive foundations: a survey. ACM Trans Appl Percept 7(1):1–39
26. Tsotsos JK (2001) Motion understanding: task-directed attention and representations that link perception with action. Int J Comput Vis 45(3):265–280

Active Vision

►Animat Vision

Activity Analysis

►Multi-camera Human Action Recognition

Activity Recognition

Wanqing Li[1], Zicheng Liu[2] and Zhengyou Zhang[3]
[1]University of Wollongong, Wollongong, NSW, Australia
[2]Microsoft Research, Microsoft Corporation, Redmond, WA, USA
[3]Microsoft Research, Redmond, WA, USA

Synonyms

Action recognition

Related Concepts

►Gesture Recognition

Definition

Activity recognition refers to the process of identifying the types of movement performed by humans over a certain period of time. It is also known as action recognition when the period of time is relatively short.

Background

The classic study on visual analysis of biological motion using moving light display (MLD) [1] has inspired tremendous interests among the computer vision researchers in the problem of recognizing human motion through visual information. The commonly used devices to capture human movement include human motion capture (MOCAP) with or without markers, multiple video camera systems, and single video camera systems. A MOCAP device usually works under controlled environment to capture the three-dimensional (3D) joint locations or angles of human bodies; multiple camera systems provide a way to reconstruct 3D body models from multiple viewpoint images. Both MOCAP and multiple camera systems have physical limitations on their use, and single camera systems are probably more practical for many applications. The latter, however, captures least visual information and, hence, is the most challenging setting for activity recognition. In the past decade, research in activity recognition has mainly focused on single camera systems. Recently, the release of commodity depth cameras, such as Microsoft Kinect Sensors, provides a new feasible and economic way to capture simultaneously two-dimensional color information and depth information of the human movement and, hence, could potentially advance the activity recognition significantly.

Regardless of which capturing device is used, a useful activity recognition system has to be independent of anthropometric differences among the individuals who perform the activities, independent of the speed at which the activities are performed, robust against varying acquisition settings and environmental conditions (for instance, different viewpoints and illuminations), scalable to a large number of activities, and capable of recognizing activities in a continuous manner. Since a human body is usually viewed as an articulated system of the rigid links or segments connected by joints, human motion can be considered as a continuous evolution of the spatial configuration of the segments or body posture, and effective representation of the body configuration and its dynamics over time has been the central to the research of human activity recognition.

Theory

Let $O = \{o_1, o_2, \cdots, o_n\}$ be a sequence of observations of the movement of a person over a period of time. The observations can be a sequence of joint angles, a sequence of color images or silhouettes, a sequence of depth maps, or a combination of them. The task of activity recognition is to label O into one of the L classes $C = \{c_1, c_2, \cdots, c_L\}$. Therefore, solutions to the problem of activity recognition are often based on machine learning and pattern recognition approaches, and an activity recognition system usually involves

extracting features from the observation sequence O, learning a classifier from training samples, and classifying O using the trained classifier. However, the spatial and temporal complexity of human activities has led researchers to cast the problem from different perspectives. Specifically, the existing techniques for activity recognition can be divided into two categories based on whether the dynamics of the activities is implicitly or explicitly modeled.

In the first category [2–9], the problem of activity recognition is cast from a temporal classification problem to a static classification one by representing activities using descriptors. A descriptor is extracted from the observation sequence O, which intends to capture both spatial and temporal information of the activity and, hence, to model the dynamics of the activity implicitly. Activity recognition is achieved by a conventional classifier such as Support Vector Machines (SVM) or K-nearest neighborhood (KNN). There are three commonly used approaches to extract activity descriptors.

The first approach builds motion energy images (MEI) and motion history images (MHI), proposed by Bobick and Davis [2], by stacking a sequence of silhouettes to capture where and how the motion is performed. Activity descriptors are extracted from the MEI and MHI. For instance, seven Hu moments were extracted in [2] to serve as action descriptors and recognition was based on the Mahalanobis distance between the moment descriptors of the trained activities and the input activity.

The second approach considers a sequence of silhouettes as a spatiotemporal volume, and an activity descriptor is computed from the volume. Typical examples are the work by Yilmaz and Shah [3] which computes the differential geometric surface properties (i.e., Gaussian curvature and mean curvature); the work by Gorelick et al. [4] which extracts space-time saliency, action dynamics, and shape structure and orientation; and the work by Mokhber et al. [5] which calculates the 3D moments of the volume.

The third approach describes an activity using a set of spatiotemporal interest points (STIPs). The general concept is first to detect STIPs from the observations O which is usually a video sequence. Features are then extracted from a local volume around each STIP, and a descriptor can be formed by simply aggregating the local features together to become a bag of features or by classifying the STIPs into a set of vocabulary (i.e., a bag of visual words) and calculating the histogram of the occurrence of the vocabulary within the observation sequence O. There are two commonly used STIP extraction techniques. One extends Harris corner detection and automatic scale selection in 2D space to 3D space and time [6] and the other is based on a pair of one-dimensional (1D) Gabor filters applied temporally and spatially [7]. Recently, another STIP detector has been proposed by decomposing an image sequence into spatial components and motion components using nonnegative matrix factorization and detecting STIPs in 2D spatial and 1D motion space using difference of Gaussian (DoG) detectors [8]. In terms of the classifier for STIP-based descriptors, besides SVM and KNN, latent topic models such as the probabilistic latent semantic analysis (pLSA) model and latent Dirichlet allocation (LDA) were used in [9]. STIP-based descriptors have a few practical advantages including being applicable to image sequences in realistic conditions, not requiring foreground/background separation or human tracking, and having the potential to deal with partial occlusions [10]. In many realistic applications, an activity may occupy only a small portion of the entire space-time volume of a video sequence. In such situations, it does not make sense to classify the entire video. Instead, one needs to locate the activity in space and time. This is commonly known as the activity detection or action detection problem. Techniques have been developed for activity detection using interest points [11].

In the second category [12–17], the proposed methods usually follow the concept that an activity is a temporal evolution of the spatial configuration of the body parts and, hence, emphasize more on the dynamics of the activities than the methods in the first category. They usually extract a sequence of feature vectors, each feature vector being extracted from a frame, or a small neighborhood, of the observation sequence O. The two commonly used approaches are temporal templates and graphical models.

The temporal-template-based approach, typically, directly represents the dynamics through exemplar sequences and adopts dynamic time warping (DTW) to compare an input sequence with the exemplar sequences. For instance, Wang and Suter [18] employed locality preserving projection (LPP) to project a sequence of silhouettes into a low-dimensional space to characterize the spatiotemporal

property of the activity and used DTW and temporal Hausdorff distance for similarity matching.

In the graphical model-based approach, both generative and discriminative models have been extensively studied for activity recognition. The most prominent generative model is the hidden Markov model (HMM), where sequences of observed features are grouped into similar configuration, i.e., states, and both the probability distribution of the observations at each state and the temporal transitional functions between these states are learned from training samples. The first work on action recognition using HMM is probably by Yamato et al. [12], where a discrete HMM is used to represent sequences over a set of vector-quantized silhouette features of tennis footage. HMM is a powerful tool to model a small number of short-term activities since a practical HMM is usually a fixed- and low-order Markov chain. Notable early extensions to overcome this drawback of the HMM are the variable-length Markov models (VLMM) and layered HMM. For details, the readers are referred to [13, 14], respectively. Recently, a more general generative graphical model, referred to as an action graph, has been established in [15], where nodes of the action graph represents salient postures that are used to characterize activities and shared by different activities, and weight between two nodes measures the transitional probability between the two postures represented by the two nodes. An activity is encoded by one or multiple paths in the action graph. Due to the sharing mechanism, the action graph can be trained and also easily expanded to new actions with a small number of training samples. In addition, the action graph does not need special nodes representing beginning and ending postures of the activities and, hence, allows continuous recognition.

The generative graphical models often rely on an assumption of statistical independence of observations to compute the joint probability of the states and the observations. This makes it hard to model the long-term contextual dependencies which is important to the recognition of activities over a long period of time. The discriminative models, such as conditional random fields (CRF), offer an effective way to model long-term dependency and compute the conditional probability that maps the observations to the motion class labels. The linear chain CRF was employed in [16] to recognize ten different human activities using features of combined shape-context and pair-wise edge features extracted at a variety of scales on the silhouettes and 3D joint angles. The results have shown that CRFs outperform the HMM and are also robust against the variability of the test sequences with respect to the training samples. More recently, Wang and Mori [17] modeled a human action by a flexible constellation of parts conditioned on image observations using hidden conditional random fields (HCRF) and achieved highly accurate frame-based action recognition.

Despite the extensive effort and progress in activity recognition research in the past decade, continuous recognition of activities under realistic conditions, such as with viewpoint invariance and large number of activities, remains challenging.

Application

Activity recognition has many potential applications. It is one of the key enabling technologies in security and surveillance for automatic monitoring of human activities in a public space and of activities of daily living of elderly people at home. Robust understanding and interpretation of human activities also allow a natural way for humans to interact with machines. A proper modeling of the spatial configuration and dynamics of human motion would enable realistic synthesis of human motion for gaming and movie industry and help train humanoid robots in a flexible and economic way. In sports, activity recognition technology has also been used in training and in the retrieval of video sequences.

References

1. Johansson G (1973) Visual perception of biological motion and a model for its analysis. Percept Psychophys 14(2): 201–211
2. Bobick A, Davis J (2001) The recognition of human movement using temporal templates. IEEE Trans Pattern Anal Mach Intell 23(3):257–267
3. Yilmaz A, Shah M (2008) A differential geometric approach to representing the human actions. Comput Vision Image Underst 109(3):335–351
4. Gorelick L, Blank M, Shechtman E, Irani M, Basri R (2007) Actions as space-time shapes. IEEE Trans Pattern Anal Mach Intell 29(12):2247–2253
5. Mokhber A, Achard C, Milgram M (2008) Recognition of human behavior by space-time silhouette characterization. Pattern Recogn 29(1):81–89

6. Laptev I, Lindeberg T (2003) Space-time interest points. In: International conference on computer vision, Nice, pp 432–439
7. Dollar P, Rabaud V, Cottrell G, Belongie S (2005) Behavior recognition via sparse-temporal features. In: 2nd joint IEEE international workshop on visual surveillance and performance evaluation of tracking and surveillance, Beijing, pp 65–72
8. Wong SF, Cipolla R (2007) Extracting spatiotemporal interest points using global information. In: International conference on computer vision, Rio de Janeiro, pp 1–8
9. Niebles JC, Wang H, Fei-Fei L (2008) Unsupervised learning of human action categories using spatial-temporal words. Int J Comput Vision 79(3):299–318
10. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos "in the wild". In: International conference on computer vision and pattern recognition (CVPR), Miami, pp 1–8
11. Yu G, Goussies NA, Yuan J, Liu Z (2011) Fast action detection via discriminative random forest voting and top-K subvolume search. IEEE Trans Multimedia 13:507–517
12. Yamato J, Ohya J, Ishii K (1992) Recognizing human action in time-sequential images using hidden Markov model. In: International conference on computer vision and pattern recognition (CVPR), Champaign, pp 379–385
13. Galata A, Johnson N, Hogg D (2001) Learning variable-length Markov models of behaviour. Comput Vision Image Underst 81:398–413
14. Oliver N, Garg A, Horvits E (2004) Layered representations for learning and inferring office activity from multiple sensory channels. Comput Vision Image Underst 96:163–180
15. Li W, Zhang Z, Liu Z (2008) Expandable data-driven graphical modeling of human actions based on salient postures. IEEE Trans Circuits Syst Video Technol 18(11): 1499–1510
16. Sminchisescu C, Kanaujia A, Metaxas D (2006) Conditional models for contextual human motion recognition. Comput Vision Image Underst 104:210–220
17. Wang Y, Mori G (2011) Hidden part models for human action recognition: probabilistic versus max margin. IEEE Trans Pattern Anal Mach Intell 33(7):1310–1323
18. Wang L, Suter D (2007) Learning and matching of dynamic shape manifolds for human action recognition. IEEE Trans Image Process 16:1646–1661

AdaBoost

Paolo Favaro[1] and Andrea Vedaldi[2]
[1]Department of Computer Science and Applied Mathematics, Universität Bern, Switzerland
[2]Oxford University, Oxford, UK

Synonyms

Adaptive boosting; Discrete adaBoost

Definition

AdaBoost is an algorithm that builds a classifier by combining additively a set of weak classifiers. The weak classifiers are incorporated sequentially one at a time so that their combination reduces the empirical exponential loss.

Background

Boosting is a procedure to combine several classifiers with weak performance into one with arbitrarily high performance [1, 2] and was originally introduced by Robert Schapire in the machine learning community [3]. AdaBoost is a popular implementation of boosting for binary classification [4]. The enthusiasm generated by boosting, and in particular by AdaBoost, in machine learning can be highlighted via a quote of Breiman [1] saying that AdaBoost with trees is the "best off-the-shelf classifier in the world." In practice, much of the popularity of AdaBoost is due to both its performance being in the same league as support vector machines [5] and its algorithmic simplicity. In the computer vision community, AdaBoost has been made very popular by the work of Viola and Jones in face detection [6]. What attracted much of the attention was that, by using a cascade of AdaBoost-trained classifiers and the notion of integral image and Haar wavelets [6, 7], Viola and Jones were able to detect faces in real time.

The boosting framework is fairly general and several implementations have been proposed, among which are AdaBoost [4]; Real AdaBoost, LogitBoost, and GentleBoost [1]; Regularized AdaBoost [8]; or extensions to multiple classes such as AdaBoost.MH [9].

Theory

This section describes the AdaBoost algorithm as originally given by Freund and Schapire [4]. This particular variant, also known as Discrete AdaBoost [1], is summarized in Algorithm 1.

The purpose of AdaBoost is to learn a *binary classifier* that is a function $H(\mathbf{x}) = y$ that maps data $\mathbf{x} \in \mathcal{X}$ (e.g., a scrap of text, an image, or a sound wave) to its

class label $y \in \{-1, +1\}$. In AdaBoost the classifier H is obtained as the sign of an additive combination of simple classifiers $h(\mathbf{x}) \in \{-1, +1\}$, called *weak hypotheses*:

$$H(\mathbf{x}) \doteq \operatorname{sign}\left(\sum_{t=1}^{m} \alpha_t h_t(\mathbf{x})\right), \tag{1}$$

where the coefficients $\alpha_t \in \mathbb{R}$. The input to AdaBoost is a set $\mathcal{H}$ of weak hypotheses and n data-label pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$; the output is a combination H of m weak hypothesis in $\mathcal{H}$ and coefficients $\alpha_1, \ldots, \alpha_m$ that fit the data, i.e., $H(\mathbf{x}_i) = y_i$ for most $i = 1, \ldots, n$.

Let us denote with H_m the classifier H with m weak hypotheses shown in Eq. (1). AdaBoost operates sequentially by adding to H_{m-1} one new weak hypothesis (h_m, α_m). While any weak hypothesis with performance better than chance can be used, it is more common to select the weak hypothesis h_m in the set $\mathcal{H}$ that minimizes the weighted *empirical error* $\epsilon(h; \mathbf{w})$, i.e.,

$$h_m = \operatorname*{argmin}_{h \in \mathcal{H}} \epsilon(h; \mathbf{w}),$$

where

$$\epsilon(h; \mathbf{w}) \doteq \frac{\sum_{i=1}^{n} w_i [y_i \neq h(\mathbf{x}_i)]}{\sum_{i=1}^{n} w_i}$$

Algorithm 1 Discrete AdaBoost

1: Initialize $F_0(\mathbf{x}) = 0$ for all $\mathbf{x} \in \mathcal{X}$.
2: Initialize $w_i = 1$ for all $i = 1, 2, \ldots, N$.
3: **for** $t = 1$ to m **do**
4: Find the weak hypothesis $h_t \in \mathcal{H}$ that minimizes

$$\epsilon(h_t; \mathbf{w}) \propto \sum_{i=1}^{N} w_i [h_t(\mathbf{x}_i) \neq y_i].$$

5: Let

$$\alpha_t \leftarrow \frac{1}{2} \log \frac{1 - \epsilon(h_t; \mathbf{w})}{\epsilon(h_t; \mathbf{w})};$$

6: Update the weights

$$w_i \leftarrow w_i e^{-y_i \alpha_t h_t(\mathbf{x}_i)};$$

7: Update the function $F_t = F_{t-1} + \alpha_t h_t$.
8: **end for**
9: Return the classifier $H(\mathbf{x}) = \operatorname{sign} F_m(\mathbf{x})$.

with positive data weights $\mathbf{w} = (w_1, \ldots, w_n)$. Here $[y_i \neq h(\mathbf{x}_i)]$ is equal to 1 if $y_i \neq h(\mathbf{x}_i)$ and 0 otherwise. Hence, the empirical error $\epsilon(h; \mathbf{w})$ is the average number of incorrect classifications of the weak hypothesis h on the weighted training data. The selected weak hypothesis h_m is then added to the current combination H_{m-1} with coefficient

$$\alpha_m = \frac{1}{2} \log \frac{1 - \epsilon(h_m; \mathbf{w})}{\epsilon(h_m; \mathbf{w})}. \tag{2}$$

While AdaBoost minimizes the empirical error of the weak hypothesis h_m at each iteration, the weights $\mathbf{w} = (w_1, \ldots, w_n)$ are chosen so that the empirical error of H_m is reduced as well. AdaBoost starts with uniform weights $\mathbf{w} = (1, \ldots, 1)$ and updates them according to the rule

$$w_i \leftarrow w_i e^{-y_i \alpha_m h_m(x_i)}, \quad i = 1, \ldots, n. \tag{3}$$

One intuitive interpretation of this rule is that it gives more importance to examples that are incorrectly classified. A formal justification is given in the next paragraph.

AdaBoost as Stagewise Minimization Denote by $F_m(\mathbf{x}) = \sum_{t=1}^{m} \alpha_t h_t(\mathbf{x})$ the additive combination of the weak hypotheses, so that the classifier $H_m(\mathbf{x})$ can be written as $\operatorname{sign} F_m(\mathbf{x})$. AdaBoost performs a stagewise minimization of the cost

$$\frac{1}{n} \sum_{i=1}^{n} e^{-y_i F_m(\mathbf{x}_i)}.$$

This cost is known as the *empirical exponential loss* and is a convex upper bound to the empirical classification error of H_m:

$$\epsilon(H_m) \doteq \frac{1}{n} \sum_{i=1}^{n} [y_i \neq H_m(\mathbf{x}_i)] \leq \frac{1}{n} \sum_{i=1}^{n} e^{-y_i F_m(\mathbf{x}_i)}.$$

To understand the effect of the AdaBoost update on the empirical exponential loss, let $F_m(\mathbf{x}) = F_{m-1}(\mathbf{x}) + \alpha_m h_m(\mathbf{x})$ be the updatedadditive combination at

iteration m. As the parameters of F_{m-1} are fixed, the empirical exponential loss is a function E of α_m and h_m:

$$E(\alpha_m, h_m) \doteq \frac{1}{n}\sum_{i=1}^{n} w_i e^{-y_i \alpha_m h_m(\mathbf{x}_i)},$$
$$\text{where } w_i = e^{-y_i F_{m-1}(\mathbf{x}_i)}. \quad (4)$$

By taking the derivative of E with respect to α_m and by setting it to zero, one obtains the optimality condition

$$0 = \sum_{i=1}^{n} w_i y_i h_m(\mathbf{x}_i) e^{-y_i h(\mathbf{x}_i)} e^{\alpha_m}$$
$$= \sum_{i:y_i \neq h_m(\mathbf{x}_i)} w_i e^{\alpha_m} - \sum_{i:y_i = h_m(\mathbf{x}_i)} w_i e^{-\alpha_m}$$

which results in the optimal coefficient given in Eq. (2):

$$\alpha_m(h_m) = \frac{1}{2}\log\frac{\sum_{i=1}^{n} w_i[y_i = h_m(\mathbf{x}_i)]}{\sum_{i=1}^{n} w_i[y_i \neq h_m(\mathbf{x}_i)]}$$
$$= \frac{1}{2}\log\frac{1-\epsilon(h_m;\mathbf{w})}{\epsilon(h_m;\mathbf{w})}.$$

By substituting this expression back in the cost (4), one obtains

$$E(\alpha_m(h_m), h_m) = 2\sqrt{\epsilon(h_m;\mathbf{w})(1-\epsilon(h_m;\mathbf{w}))}\sum_{i=1}^{n} w_i$$

which achieves its smallest value when the empirical classification error $\epsilon(h_m;\mathbf{w})$ approaches either 0, its minimum, or 1, its maximum. Notice that if the error $\epsilon(h_m;\mathbf{w}) > 1/2$, then the corresponding weight α_m is negative. In other words, when the weak hypothesis h_m makes more mistakes than correct classifications, AdaBoost automatically swaps the sign of the output label so that $\epsilon(-h_m;\mathbf{w}) < 1/2$. Finally, the weight update Eq. (3) follows from

$$w_i \leftarrow e^{-y_i F_m(\mathbf{x}_i)} = e^{-yF_{m-1}(\mathbf{x}_i)} e^{-y_i \alpha_m h_m(\mathbf{x}_i)}$$
$$= w_i e^{-y_i \alpha_m h_m(\mathbf{x}_i)}.$$

Application

One of the main uses of AdaBoost is for the recognition of patterns in data. Recognition can be formulated as a binary classification problem: Find whether data points match the pattern of interest or not. In computer vision, AdaBoost was popularized by its application to object detection, where the task is not only to recognize but also to localize within an image an object of interest (e.g., a face). Most of the ideas summarized in this section where first proposed by Viola and Jones [6].

A common technique for object detection is the *sliding window* detector. This method reduces the object detection problem to the task of classifying all possible image windows (i.e., patches) to find which ones are centered around the object of interest. In practice, windows may be sampled not only at all spatial locations but also at all scales and rotations. This results in a very large number of evaluations of the classifier function for each input image. Therefore, the computational efficiency of the classifier is of paramount importance.

Classifiers computed with AdaBoost can be made very computationally efficient by using weak hypotheses that are fast to compute and by letting AdaBoost select a small set of hypotheses most useful to the given problem. For example, in the Viola-Jones face detector, a weak hypothesis is computed by thresholding the output of a linear filter that computes averages over rectangular areas of the image. These filters are known as Haar wavelets and, because of their special structure, can be computed in constant time by using the *integral image* [6].

In order to further improve the speed of a sliding window detector, AdaBoost classifiers are often combined in a *cascade* [6]. A cascade exploits the fact that the vast majority of image windows are *not* centered around the object of interest and that, furthermore, most of these negative windows are easy to recognize as such. A cascade is built by appending one AdaBoost classifier after another. Classifiers are evaluated sequentially and an image window is rejected as soon as the response of a classifier is negative. All the classifiers are tuned to almost never reject a window that matches the object of interest (i.e., high recall). However, the first classifiers in the cascade are allowed to return several false positives (i.e., low precision) in exchange for a significantly reduced evaluation cost, obtained, for instance, by limiting the number of weak

hypotheses in them. By using this scheme, the computationally costly and highly accurate classifiers are evaluated only on the most challenging cases: Windows that resemble the object of interest and that therefore either contain the object (i.e., a positive sample) or a visual structure that can be easily confused with it (i.e., a hard negative sample).

Finally, since each weak hypothesis is usually associated to an elementary feature, AdaBoost is also often used for *feature selection*. In some cases, feature selection improves the *interpretability* of the classifier. For instance, in the Viola-Jones face detector, the first few Haar wavelets selected by AdaBoost usually capture semantically meaningful anatomical structures such as the eyes and the nose.

References

1. Friedman JH, Hastie T, Tibshirani R (2000) Additive logistic regression: a statistical view of boosting. Ann Stat 28(2): 337–374
2. Hastie T, Tibshirani R, Friedman J (2001) The elements of statistical learning. Springer, New York
3. Schapire RE (1990) The strength of weak learnability. Mach Learn 5(2):197–227
4. Freund Y, Schapire RE (1996) Experiments with a new boosting algorithm. In: Proceedings of the 13th international conference on machine learning, Bari, 148–156
5. Vapnik V (1995) The nature of statistical learning theory. Springer, New York
6. Viola P, Jones M (2001) Robust real-time object detection. In: Proceedings of IEEE workshop on statistical and computational theories of vision, Vancouver, Canada
7. Papageorgiou CP, Oren M, Poggio T (1998) A general framework for object detection. In: International conference on computer vision, Bombay, pp 555–562
8. Sun Y, Li J, Hager W (2004) Two new regularized adaboost algorithms. In: Machine learning and applications, Louisville, pp 41–48
9. Schapire RE, Singer Y (1998) Improved boosting algorithms using confidence-rated predictions. In: Computational learning theory, Springer, New York, pp 80–91

Adaptive Boosting

▸AdaBoost

Adaptive Gains

▸von Kries Hypothesis

Affine Alignment

▸Affine Registration

Affine Camera

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Synonyms

Affine projection

Related Concepts

▸Perspective Camera; ▸Perspective Transformation; ▸Weak Perspective Projection

Definition

An *affine camera* is a linear mathematical model to approximate the perspective projection followed by an ideal pinhole camera.

Background

Perspective projections give accurate models for a wide range of existing cameras, especially after calibration for a limited volume of workspace. However, the relationship from a 3D point to its 2D image point is nonlinear due to a scalar factor dependent of each individual point (see entry "▸Perspective Camera" for details). *Affine cameras* are introduced to make the projection model more mathematically tractable. An affine camera model is a first-order approximation obtained from Taylor expansion of the perspective camera model around a reference point. The reference point can be any point, but it may be set to the centroid of the 3D points, which results in a more accurate approximation. There are three important instances of an affine camera when the camera's intrinsic parameters are known: orthographic, weak perspective, and paraperspective projections. The reader is

referred to entry "▸Weak Perspective Projection" for a detailed description of the affine camera model and its three instances.

Affine Invariants

Michael Werman
The Institute of Computer Science, The Hebrew University of Jerusalem, Jerusalem, Israel

Concept

Images of even the same object undergo various transformations depending on changes in the camera and its settings, the lighting, and the object itself. One of the ways to handle some of these changes, for tracking, search, and understanding images, is to use a description of the object that is oblivious to some of the above-mentioned transformations. Affine invariants are commonly used for this purpose.

There is a vast literature relating to affine invariants and only a small selection will be mentioned [4, 5, 11, 14, 17].

Affine Transformation

In order to have a property of an object that is invariant to an affine transformation, affine invariants can be used.

$$x \Rightarrow Lx + t$$

is an affine transformation where $x \in R^n$ is a vector, $L \in R^{n \times n}$ a matrix, and $t \in R^n$ a vector. L is a linear transformation, and t is a translation [2].

Affine transformations are used to describe different changes that images can undergo, such as an affine transformation of the (r, g, b) color values of an object under different lighting conditions or the transformation the shape of the image of an object undergoes when the camera and object are in different relative positions. The affine transformation in these cases does not necessarily model the exact physical distortion that is seen but often is a good approximation.

In order to have a property of an object that is invariant to an affine transformation, affine invariants can be used.

Affine Invariant

Let f be a function such that $f(a, b, \ldots) = f(T(a), T(b), \ldots)$ for any affine transformation T and $a, b, \ldots \in R^n$, f is an affine invariant.

First example Let p, q, r be real numbers then an affine transformation is of the form $x \Rightarrow \alpha x + \beta$; it is easy to check that for any three numbers p, q, r, and affine transformation parameters α, β,

$$\frac{p-q}{p-r} = \frac{(\alpha p + \beta) - (\alpha q + \beta)}{(\alpha p + \beta) - (\alpha r + \beta)}$$

[16, 18]

Second example We can generalize the first example to any dimension, d. The ratio of the volume of any two sets is an affine invariant (constant Jacobian, $|L|$) and is proportional to the $d + 1 \times d + 1$ determinant:

$$\begin{vmatrix} p & q & \ldots & r \\ 1 & 1 & \ldots & 1 \end{vmatrix}$$

Algebraic curves Not only points can be used to define affine invariance; other common examples are the parameters of curves, such as the equations of lines, conics, or other algebraic curves. For degree two curves in the plane, all ellipses are affinely equivalent.

Affine differential geometry There are affine invariant analogues of arc length and curvature. In general it is possible to find affine invariants involving points and their derivatives [3, 10, 13, 15].

Other parameters of the object Fourier coefficients and moments.

An affine transformation has $d^2 + d$ parameters, so in order to have a nontrivial affine invariant, one needs a function with more than that many arguments, where the result can be computed using algebraic elimination [9]. For example, two simplices in R^d have at least $d + 2$ different points which is $d^2 + 2d$ arguments.

In a certain sense, the number of independent invariants is the #(of parameters of a configuration) $-(d^2 + d)$.

There are other properties that are affine invariant, such as incidence, parallelism, centroids, barycentric coordinates, convexity, tangency, bi-tangency, Euler number, and connectivity.

If an objects area is A, then integrating by the area element divided by A is affine invariant, for example, $\frac{1}{A}\int_{Object} g(I(x,y))dxdy$ is affine invariant, g being any function of the pixel's color.

Affine Invariant Feature Detection

There are a number of affine invariant feature detectors that find affine invariant local features in an image [7, 17]. One of the successes in recent practice of computer vision was the SIFT, a similarity invariant feature, and its extension to an affine invariant feature, ASIFT [6, 8].

Normalization

There are normalizations that can be done to an object that remove all/some of the possible affine variation.

The center of gravity is a linear invariant, so it is possible to translate the object so that the object's center is at a fixed coordinate, thus removing the translation term, changing the problem from one of finding an affine invariant to a linear invariant.

The Whitening transform canonically transforms a set of points using an affine transformation so that the average is 0 and the covariance matrix is the identity, I.

Grassmannians

The set of labeled points modulo affine transformations are isomorphic to Grassmannians, using this one can define a geometry of affine invariant point sets and, for example, measure the distance between *affine invariant point sets* [1].

Correspondence

Another thing to notice is that usually there needs to be some correspondence of the objects in order to use these invariants, namely, the order of the arguments of the function f needs to be known, and the way to overcome this problem is summing over all permutations making this function invariant both to permutations. Another possibility is using permutation invariant features, for example, moments, which are built from summing over all the points.

Noise

Even though the group of affine transformation may be only a subgroup of the possible transformations, for example, projective transformations for an image of planar scene, being affinely invariant maybe an overkill as most of the radical transformations never really happen. Thus, noise can be explained by affine transformations making too many things close to each other.

References

1. Begelfor E, Werman M (2006) Affine invariance revisited. In: IEEE conference on computer vision pattern recognition (CVPR), New York
2. Gallier JH (2001) Geometric methods and applications for computer science and engineering. Springer, New York
3. Gotsman C, Werman M (1993) Recognition of affine transformed planar curves by extremal geometric properties. Int J Comput Geom Appl 3(2):183–202
4. Govindu VM, Werman M (2004) On using priors in affine matching. Image Vis Comput 22(14):1157–1164
5. Hartley RI, Zisserman A (2004) Multiple view geometry in computer vision, 2nd edn. Cambridge University Press, Cambridge/New York. ISBN:0521540518
6. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60:91–110
7. Mikolajczyk K, Tuytelaars T, Schmid C, Zisserman A, Matas J, Schaffalitzky F, Kadir T, Van Gool L (2005) A comparison of affine region detectors. Int J Comput Vis 65(1):43–72
8. Morel J-M, Yu G (2009) Asift: a new framework for fully affine invariant image comparison. SIAM J Imaging Sci 2(2):438–469
9. Olver PJ (2009) Equivalence, invariants and symmetry. Cambridge University Press, Cambridge
10. Olver PJ (2009) Moving frames and differential invariants in centroaffine geometry. Technical report
11. Petrou M, Kadyrov A (2004) Affine invariant features from the trace transform. IEEE Trans Pattern Anal Mach Intell 26:30–44
12. Rahtu E, Salo M, Heikkil J (2005) Affine invariant pattern recognition using multiscale autoconvolution. IEEE Trans Pattern Anal Mach Intell 27:908–918
13. Reid M, Szendroi B (2005) Geometry and topology. Cambridge University Press, Cambridge/New York
14. Sapiro G (2000) Geometric partial differential equations and image analysis. Cambridge University Press, Cambridge

15. Simon U (2000) Affine differential geometry [Chapter 9]. In: Handbook of differential geometry, vol 1. North-Holland, pp 905–961
16. Sprinzak J, Werman M (1994) Affine point matching. Pattern Recognit Lett 15(4):337–339
17. Tuytelaars T, Mikolajczyk K (2008) Local invariant feature detectors: a survey. Found Trends Comput Graph Vis 3: 177–280
18. Werman M, Weinshall D (1995) Similarity and affine invariant distances between 2D point sets. IEEE Trans Pattern Anal Mach Intell 17(8):810–814

Affine Projection

Zhengyou Zhang
Microsoft Research, Redmond, WA, USA

Synonyms

Affine camera

Related Concepts

▸Affine Camera; ▸Perspective Camera; ▸Perspective Transformation; ▸Weak Perspective Projection

Definition

An *affine projection* is a linear mathematical model to describe the projection performed by an affine camera. See entry "▸Affine Camera" for details.

Affine Registration

Kevin Köser
Institute for Visual Computing, ETH Zurich, Zürich, Switzerland

Synonyms

Affine alignment

Related Concepts

▸Affine Camera; ▸Rigid Registration

Definition

The goal of affine registration is to find the affine transformation that best maps one data set (e.g., image, set of points) onto another.

Background

In many situations data is acquired at different times, in different coordinate systems, or from different sensors. Such data can include sparse sets of points and images both in 2D and 3D, but the concepts generalize also to higher dimensions and other primitives. *Registration* means to bring these data sets into alignment, i.e., to find the "best" transformation that maps one set of data onto another, here using an *affine transformation*. For the sake of brevity, in this entry only the registration of two data sets is discussed, although approaches exist for finding consistent transformations that align more than two sets at once (e.g., [1]). While intuitively in 1D affine transformations compensate for scale and offset, in any dimension they can represent the first-order Taylor approximation (local linearization) for nonlinear functions.

For more complicated transformations, often a first affine registration is used as an initial solution that roughly aligns the data sets, followed by some (potentially local or nonlinear) search for more complicated parameters. In contrast to rigid registration, which allows only an offset and a rotation between the data sets, the affine model allows also for the full set of linear shape changes, including (nonisotopic) scale and shear. In particular, locally this approximates perspective effects or other nonlinear warps. On the other hand, affine registration uses still a single global transformation with a set of a few global parameters, which makes it mathematically easier to handle but also less powerful as compared to general nonrigid registration of deformable or articulated objects.

Theory

Two important cases can be distinguished for registration, the purely geometrical case, where two (finite) sets of points have to be registered, or the continuous/functional case, where the similarity of two functions must be maximized by an affine transformation of the functions' domain.

The Purely Geometrical Case

Let X and Y be two sets of points from $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively, and without loss of generality, it is assumed that $m \geq n$. For now it is assumed that the sets are of the same size and that there exists an (unknown) affine transformation, such that each element in X is mapped to an element in Y. Then, the goal of affine registration is to find this transformation, i.e., the matrix $A \in \mathbb{R}^{n\times m}$ and the offset $t \in \mathbb{R}^n$, such as to minimize an energy:

$$E_d = \sum_{x\in X} \min_{y\in Y} d_g(Ax + t, y) \tag{1}$$

Here $d_g(a, b)$ encodes the distance between a and b. Usually the points in X or Y are not available directly but only their noisy observations (e.g., when y are observed 2D projections of known 3D points and the parameters of the affine camera [2] are sought). In that case affine registration is based on these observations, and d_g should be chosen according to the noise distribution. If the sets are not of the same size or not each point in X corresponds to a point in Y, or in case there are gross errors in the data, still E_d can be minimized, e.g., with a robust cost function d_g to obtain some alignment between the sets [3]. Intuitively, in all these cases, A and t are sought which minimize the average distance for a transformed point of X to the closest point in Y.

In case bounds on A and t are known, a simple way to find an approximate solution is to sample the (continuous) space of all possible matrices A and offsets t, but this is usually computationally intractable. If it is known which points in X correspond to which others in Y, i.e., there is a set of pairs (x_i, y_i), the energy can be rewritten as

$$E_c = \sum_{x_i\in X} d_g(Ax_i + t, y_i) \tag{2}$$

In case $d_g(a, b) = ||(a - b)||_2^2$, this can be solved explicitly. A contains mn unknowns, t contains n unknowns, and from each correspondence n observations can be obtained, so from counting it is clear that at least $m+1$ correspondences are required to uniquely determine a solution (assuming they are in a general configuration and not, e.g., only on a line). In case the correspondences contain gross errors (mismatches, outliers), then the solution can be obtained using robust estimation techniques such as least median of squares [4] or RANSAC [5].

In case correspondences are not known, but that the data sets are already approximately aligned, it can be assumed that the closest neighbors are in correspondence. In this case the problem can be solved by using the iterative closest points (ICP) method [6–8]. Locally, for each point the offset to the closest point in the other data set is computed, and by collecting all the local offsets, a global affine transformation is estimated that aligns these sets (in a least squares sense according to Eq. 2). This transformation is applied to recalculate the nearest neighbor correspondences and the estimation step is repeated with these. In [9] efficient implementations and practical aspects of ICP have been studied. However, since ICP relies on local neighborhoods, it cannot cope well with situations where initially close points must be moved far away.

A technique proposed for this is based on normalization (as suggested, e.g., by [10]): Here, the means of the data sets are computed individually and t is defined as the difference of the means. Then, for each data set the (unbiased) covariance is computed and the matrices A_X,A_Y are searched that bring the respective point distributions to a unit covariance matrix (whitening of covariance). The two data sets now only differ by an orthogonal matrix that may be obtained by sampling or finding other characteristics in the data (cf. also [11]). However, this normalization approach assumes that the data sets overlap fully.

In general, similar concepts can be applied as for rigid registration, however, with a slightly different set of parameters.

The Continuous/Functional Case

Let I and J be functions (images) from $\mathbb{R}^n$ to $\mathbb{R}^d$, assigning some color value to each position, typically in the plane or in space. The function value will be referred to as the color hereafter, regardless of its

physical meaning. In this case the affine registration can be stated as the minimization of an energy:

$$E = \int_x d_c(I(Ax + t), J(x))dx \tag{3}$$

Here, d_c is a distance between two colors that should be chosen according to the expected measurement uncertainty of the colors. Very similar to the discrete case, a naive strategy to minimize this energy would be sampling; however, this is again computationally expensive and requires bounds on the parameters.

A possible solution is to compute local image features, such as corners, blobs, and so on (cf. to [12]), and – if possible – find correspondences among these features. There has been a huge body of work to particularly define features that can be detected reliably with affine changes of image coordinates. A comparison can be found in [13]. Given those the discrete methods of the previous section can be used to find an alignment A, t.

Afterwards, or in case A and t were approximately known from the beginning and if I and J are smooth, then the alignment can be performed by local linearization as proposed by Lucas and Kanade [14]. The assumption is that locally the image can be represented by its first-order Taylor approximation, i.e., the local color and the gradient:

$$I(x) \approx I(x_0) + \frac{\partial I}{\partial x} \underbrace{(x_0 - x)}_{\Delta x} \tag{4}$$

Consequently, given two almost aligned images, at position x_0

$$\underbrace{J(x_0) - I(x_0)}_{\Delta I_0} \approx \left.\frac{\partial I}{\partial x}\right|_{x_0} \Delta x \tag{5}$$

Stacking r of these equations on top of each other, an equation system is obtained:

$$\begin{pmatrix} \Delta I_0 \\ \Delta I_1 \\ \cdots \\ \Delta I_{r-1} \end{pmatrix} \approx \begin{pmatrix} \left.\frac{\partial I}{\partial x}\right|_{x_0} \\ \left.\frac{\partial I}{\partial x}\right|_{x_1} \\ \cdots \\ \left.\frac{\partial I}{\partial x}\right|_{x_{r-1}} \end{pmatrix} \Delta x \tag{6}$$

This can be solved in a least squares sense to obtain Δx.

Similar to the local image gradient with respect to position, also the partial derivatives with respect to affine transformation parameters can be computed [14]. In this case, Δx (and also ∂x) contains $n(n + 1)$ parameters (n^2 for the linear shape change and n for the offset). At least $n(n + 1)$ equations are required to uniquely solve this. Often, to increase the basin of convergence, coarse-to-fine registration is applied. This can be implemented using image pyramids in case the uncertainty lies mostly in the offset parameters or by propagation of the affine parameter uncertainty to position uncertainty in the image and appropriate smoothing [15]. After having applied the estimated update Δx on the parameters, the steps can be applied repeatedly to register the two images. In [16] Baker and Matthews compare different formulations of such iterative, gradient-based image alignment, particularly the question of how to compose and parameterize the warps across multiple iterations.

On top of transformations on the domain of the images, often the two images differ in target (e.g., the colors of corresponding positions are related by some brightness offset), in which case also parameters for the change of color need to be estimated. In case the corresponding image colors are only statistically related but no explicit transformation model between colors is known, the concept of mutual information [17] might be used, where the entropy of the joint color histogram is minimized. An overview of image-based alignment can be found in [18, 19].

Application

Affine cameras [2] approximate a real camera by an affine mapping of 3D points to 2D image coordinates. For tracking local regions through videos, it has been shown [20] that keeping track of the affine deformations of local regions can help detecting tracking failures. Such affine warps, or those implied by correspondences of affine features between different images, represent the local linearization of a potentially nonlinear image warp (e.g., of perspective effects). If the structure of this nonlinear warp is known, the affine registration can allow inferring the global warp directly [21] or provide more constraints than just using the position of a region correspondence.

In general, affine registrations often provide a reasonable solution to align mean, linear shape, and orientation of data without making the transformation too problem specific, or the affine solution can serve as a basis for further more advanced alignment. Furthermore, multiple independent or coupled local affine registrations can help in registering articulated or deformable models.

References

1. Eggert DW, Fitzgibbon AW, Fisher RB (1998) Simultaneous registration of multiple range views for use in reverse engineering of CAD models. Comput Vis Image Underst 69(3): 253–272
2. Mundy JL, Zisserman A (eds) (1992) Geometric invariance in computer vision. MIT, Cambridge
3. Fitzgibbon AW (2003) Robust registration of 2d and 3d point sets. Image and Vis Computing 21:1145–1153
4. Rousseeuw PJ (1984) Least median of squares regression. J Am Stat Assoc 79(388):871–880
5. Fischler M, Bolles R (1981) RANdom SAmpling Consensus: a paradigm for model fitting with application to image analysis and automated cartography. Commun ACM 24(6):381–395
6. Besl PJ, McKay ND (1992) A method for registration of 3-d shapes. IEEE Trans Pattern Anal Mach Intell 14:239–256
7. Zhang Z (1994) Iterative point matching for registration of free-form curves and surfaces. Int J Comput Vis 13:119–152
8. Feldmar J, Ayache N (1994) Rigid and affine registration of smooth surfaces using differential properties. In: Eklundh JO (ed) European conference on computer vision (ECCV '94). Volume 801 of lecture notes in computer science. Springer, Berlin/Heidelberg, pp 396–406. doi:10.1007/BFb0028371
9. Rusinkiewicz S, Levoy M (2001) 3-D digital imaging and modeling. Proceedings Third International Conference on, Efficient variants of the ICP algorithm, 145–152. doi:10.1109/IM.2001.924423
10. Obdrzálek S, Matas J (2002) Local affine frames for image retrieval. Proceedings of the International Conference on Image and Video Retrieval, CIVR '02. Springer, London, pp 318–327
11. Ho J, Yang MH (2011) On affine registration of planar point sets using complex numbers. Comput Vis Image Underst 115(1):50–58
12. Tuytelaars T, Mikolajczyk K (2008) Local invariant feature detectors: A survey. Found Trends Comput Graph Vis 3(3):177–280
13. Mikolajczyk K, Tuytelaars T, Schmid C, Zisserman A, Matas J, Schaffalitzky F, Kadir T, van Gool L (2005) A comparison of affine region detectors. Int J Comput Vis 65(1–2):43–72
14. Lucas BD, Kanade T (1981) An iterative image registration technique with an application to stereo vision. Proceedings of the 7th international joint conference on Artificial intelligence, 2(6):674–679
15. Köser K, Koch R (2008) Exploiting uncertainty propagation in gradient-based image registration. Proc Br Mach Vis Conf 83–92
16. Baker S, Matthews I (2004) Lucas-kanade 20 years on: A unifying framework. Int J Comput Vis 56(1): 221–255
17. Viola PA III, WMW (1997) Alignment by maximization of mutual information. Int J Comput Vis 24(2):137–154
18. Zitov B, Flusser J (2003) Image registration methods: a survey. Image Vis Comput 21:977–1000
19. Szeliski R (2006) Image alignment and stitching: A tutorial. Found Trends Comput Graph Comput Vis 2
20. J Shi, Tomasi C (1994) Good features to track. IEEE Conf Comput Vis Pattern Recognit 593–600
21. Köser K, Koch R (2008) Differential spatial resection – pose estimation using a single local image feature. Eur Conf 278 Comput Vis (LNCS 5302Ű5305), 312–325

Affordances

▸Affordances and Action Recognition

Affordances and Action Recognition

James Bonaiuto
California Institute of Technology, Pasadena, CA, USA

Synonyms

Action Recognition; Affordances

Related Concepts

▸Gait Recognition; ▸Gesture Recognition; ▸Multi-camera Human Action Recognition

Definition

Affordances are opportunities for action that are directly perceivable in an organism's environment without higher-level cognitive functions. Action recognition is the result of mapping an observed action onto an internal motor or semantic representation.

Background

Affordances are defined by Gibson [1] as opportunities for action that are directly perceivable without the need for higher-level cognitive functions such as object recognition. The concept of affordances for action has generated significant interest in the computer vision and robotics community. More recently, links between this concept and that of action recognition have been explored, suggesting that the two may share common mechanisms.

Affordances. In robotics, early use of the term affordances dealt with the extraction of features from the visual environment that signal the possibility for action. However, neural models of vision and action have suggested a more active role for affordances in monitoring the ongoing control of an action and in predicting the effects of an action on a target object [2–4].

Because the affordances available to an agent depend on its embodiment, their representation needs to be grounded in its perceptual and motor repertoire. Most approaches therefore learn affordances through a stage of motor babbling with objects, where different actions are randomly attempted on an object while recording the initial perception of the object and any effects that the action has on it.

Typically, affordance formalisms involve storing entity, action, and effects relations, allowing the agent to associate objects with possible actions to perform on them and the effects of these actions. This formulation is therefore useful for planning by allowing an agent to search for an action that will produce a desired effect. More powerful formalisms store relations between entity, action, and effect equivalence classes [5]. These allow an agent to filter incoming sensory data to extract invariant affordance cues and to generalize learned affordances to novel objects.

Action Recognition. Action recognition can take place on several hierarchical scales of organization. At the lowest level, this consists of recognizing basic actions like reaching and grasping. Recognition of these actions can be used to recognize more complex actions and sequences of actions such as putting sugar in a cup. The highest level of action recognition involves inferring the goals, intentions, or the task of the observed agent (e.g., making coffee).

The two main approaches to low- and mid-level action recognition in computer vision are model-based and template-based or holistic. Model-based action recognition involves tracking body parts using a parametric model of the body kinematics. Body parts are typically recognized by low-level features, and then geometric constraints from the body model are imposed on them. Actions are typically represented in joint space. Template-based action recognition directly models actions using spatiotemporal features and represents actions as spatio-temporal shapes in 3 or 4d space. These features could be static features based on edges and limb shapes, dynamic features based on optical flows, or space-time volumes.

Neural models of low-level action recognition commonly use self-observation during action performance to associate the motor representation used to generate an action with the visual representation elicited by observation of it [6, 7]. An object-centered representation is hypothesized to exist for feedback-based control of actions, which causes self-observation during performance of an action and observation of another agent performing the same action to elicit similar visual representations [6, 8]. This link between affordance-activated motor representations and visual representations of observed actions is thought to ground understanding of the goals and intentions of other agents.

In order to recognize more complex actions and action sequences, the results of low-level action recognition are typically combined using some sequence parsing technique. Previous approaches have used hidden Markov models (HMMs) [9], probabilistic parsing using context-free grammars [10], and Bayesian inference on graphical models [11]. Models of action recognition at the highest level use lower level motor mechanisms in a simulation mode to infer the intention of the observed actor [12] or Bayesian inverse reinforcement learning to infer the reward function, or task, of a demonstration [13].

Theory

The close relationship between affordance and action recognition is made apparent by the related ideas that affordances play an active role in online control of an action and that the effects of an action on an object are included in the affordance representation. The same mechanisms used to monitor and control an agent's ongoing action can therefore also be leveraged to recognize the same action performed

by another agent. Mechanisms for affordance perception and action recognition can interact through several ways. Recognition of an action performed by another agent can allow an agent to learn affordance cues or the effects of an affordance by observation. Observation of the effects of an affordance can aid in action recognition when the action itself appears ambiguous. Finally, extraction of affordance cues can similarly aid in action recognition by narrowing the space of possible actions.

Several systems use previously learned affordance information to recognize observed actions by searching for an action that is afforded by the observed object [14] or that would have produced the observed effect on the object [11, 13]. In both cases, the agent interprets the observed actions of other agents in terms of its own affordance representations. This can improve recognition accuracy when the details of the motion of the body parts during a particular is too difficult to disambiguate, but the effects of the action on the object are clearly perceivable.

Given a system for recognizing low-level body part movements, the affordances of objects can be approximately learned by observation [15]. This requires that the robot have a similar embodiment to that of the observed agent or a mechanism to map from observed body parts onto its own body. In this case, the features that indicate an affordance and the effects of the observed action can easily be recorded. If the observed action is novel and cannot be recognized at a high level, the recognized low-level subcomponents can be used to guide subsequent attempts at reproducing it by trying to replicate its effects on an object.

References

1. Gibson JJ (1966) The senses considered as perceptual systems. Houghton Mifflin, Boston
2. Fagg AH, Arbib MA (1998) Modeling parietal-premotor interactions in primate control of grasping. Neural Netw 11(7–8):1277–1303
3. Paletta L, Fritz G, Kintzler F, Irran J, Dorffner G (2007) Learning to perceive affordances in a framework of developmental embodied cognition. IEEE International Conference on Development and Learning, London, pp 110–115
4. Arbib MA (1997) From visual affordances in monkey parietal cortex to hippocampoparietal interactions underlying rat navigation. Phil Trans R Soc Lond B 352(1360): 1429–1436
5. Sahin E, Cakmak M, Dogar MR, Ugur E, Ucoluk G (2007) To afford or not to afford: a new formalization of affordances toward affordance-based robot control. Adapt Behav 15(4):447–472
6. Oztop E, Arbib MA (2002) Schema design and implementation of the grasp-related mirror neuron system. Biol Cybern 87(2):116–140
7. Metta G, Sandini G, Natale L, Craighero L, Fadiga L (2006) Understanding mirror neurons: a bio-robotic approach. Interact Stud 7(2):197–232
8. Bonaiuto J, Rosta E, Arbib MA (2007) Extending the mirror neuron system model, I: audible actions and invisible grasps. Biol Cybern 96:9–38
9. Yamato J, Ohya J, Ishii K (1992) Recognizing human action in time-sequential images using hidden Markov model. In: Proceedings of computer vision and pattern recognition (CVPR), Champaign, IL, pp 379–385
10. Bobick AF, Ivanov YA (1998) Action recognition using probabilistic parsing. In: Proceedings of computer vision and pattern recognition (CVPR), Santa Barbara, pp 196–202
11. Gupta A, Davis LS (2007) Objects in action: an approach for combining action understanding and object perception. In: Proceedings of computer vision and pattern recognition (CVPR), Minneapolis, pp 1–8
12. Oztop E, Wolpert D, Kawato M (2005) Mental state inference using visual control parameters. Cogn Brain Res 22:129–151
13. Lopes M, Melo FS, Montesano L (2007) Affordance-based imitation learning in robots. In: IEEE/RSJ international conference on intelligent robots and systems, San Diego, CA, pp 1015–1021
14. Moore DJ, Essa IA, Hayes MH (1999) Exploiting human actions and object context for recognition tasks. Proc Int Conf Comput Vis (ICCV) 1:80–86
15. Kjellstrom H, Romeroa J, Kragic D (2011) Visual object-action recognition: inferring object affordances from human demonstration. Comput Vis Image Underst 115(1): 81–90

Algebraic Curve

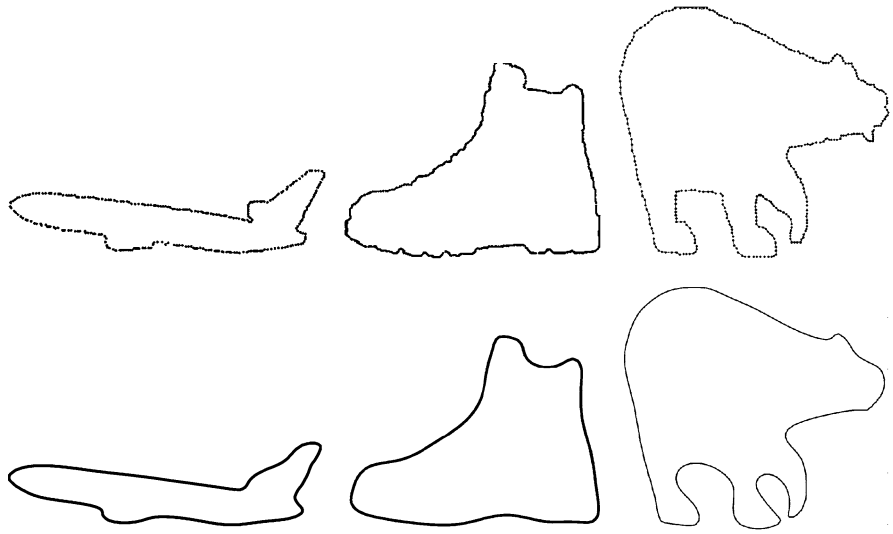
Bo Zheng
Computer Vision Laboratory, Institute of Industrial Science, The University of Tokyo, Meguro-ku, Tokyo, Japan

Synonyms

Implicit polynomial curve

Related Concepts

▸Algebraic Surface



Algebraic Curve, Fig. 1
Examples of algebraic curves. *Top row*: original data sets; *bottom row*: represented algebraic curves

Definition

An *algebraic curve* is a curve determined by a 2-D *implicit polynomial* (IP) of degree n:

$$f_n(\mathbf{x}) = \sum_{0 \le i,j; i+j \le n} a_{ij} x^i y^j = (\underbrace{1 \; x \; \ldots \; y^n}_{\mathbf{m}^{\mathrm{T}}})(\underbrace{a_{00} \; a_{10} \; \ldots \; a_{0n}}_{\mathbf{a}})^{\mathrm{T}} = 0, \quad (1)$$

where $\mathbf{x} = (x, y)^{\mathrm{T}}$ is the coordinate of a point on a curve. That is, the curve is always represented by f_n's zero level set: $\{\mathbf{x} | f_n(\mathbf{x}) = 0\}$. The polynomial function is usually denoted by an inner product between two vectors: monomial vector $\mathbf{m}$ and coefficient vector $\mathbf{a}$. For the entries in these vectors, indices $\{i, j\}$ can be arranged in different orders, such as *lexicographical order* or *inverse lexicographical order*. In addition, the *homogeneous binary polynomial* of degree r in x and y, $\sum_{i+j=r} a_{ij} x^i y^j$, is called the r-th degree form of the IP. The form of degree n is called the *leading form*. The degree of an algebraic curve is the degree of the polynomial (e.g., n). An algebraic curve of degree 2 is called a conic, degree 3 a cubic, degree 4 a quartic, and so on.

Background

In computer vision, representing 2-D data sets with algebraic curves has been studied extensively for the past three decades. It is attractive for vision applications due to its applicability to object recognition, pose estimation, and registration. In contrast to the curves represented by other functions such as splines, Fourier, rational Gaussian, and radial basis function, algebraic curve is superior in such areas as fast fitting, few parameters, algebraic/geometric invariants, and robustness against noise and occlusion. Algebraic curve is also capable of modeling non-star shapes, open curves, curves that contain gaps, and unordered curve data. However, algebraic curve representation still suffers from some major issues such as the lack of local accuracy and global stability when representing a complex 2-D shape (see [7]). Figure 1 shows some example algebraic curves successfully used to represent closed 2-D curves.

Application and Theory

Algebraic curve representation is mainly attractive for vision applications such as fast shape registration or pose estimation [3, 6, 8, 9, 12] and recognition [2, 4–6, 9–11]. To achieve these purposes, many efforts have been made in three topics: curve fitting, algebraic/geometric invariants, and curve registration. The first is about solving the problem of accurately and stably fitting an algebraic curve to a complex shape, the second is on extracting algebraic or geometric invariants from the algebraic curve representing a shape, and the third concerns estimating

the Euclidean transformation(s) between two or more algebraic curves representing different instances of the same shape.

Curve Fitting. There have been great improvements concerning algebraic curve fitting with its increased use during the late 1980s and early 1990s [8]. Recently, new robust and consistent fitting methods like 3L fitting [1], gradient-one fitting with Rigid regression [7], and degree-adaptive fitting [13] that are suitable for vision applications have been introduced.

Algebraic/Geometric Invariants. The main advantage of the use of algebraic curves for recognition is the existence of algebraic/geometric invariants, which are functions of the polynomial coefficients that do not change after a coordinate transformation. To some major contributions, the global Euclidean invariants are found by Taubin and Cooper [9], Teral and Cooper [6], and Keren [2], which can be expressed as simple explicit functions of the IP coefficients. Wolovich et al. [11] also introduced a set of invariants from covariant conic decompositions of implicit polynomials.

Curve Registration. In prior literatures [6, 9], global shape registration is performed through a single (non-iterative) computation using the central and oriented information extracted from the polynomial coefficients of two algebraic curves. Recently, an iterative method for aligning partially matched curves that uses the distance measurement of the polynomial gradient field together with a fast polynomial transformation has been introduced [12].

References

1. Blane M, Lei ZB, Cooper DB (2000) The 3L algorithm for fitting implicit polynomial curves and surfaces to Data. IEEE Trans Pattern Anal Mach Intell 22(3):298–313
2. Keren D (1994) Using symbolic computation to find algebraic invariants. IEEE Trans Pattern Anal Mach Intell 16(11):1143–1149
3. Keren D, Cooper D, Subrahmonia J (1994) Describing complicated objects by implicit polynomials. IEEE Trans Pattern Anal Mach Intell 16(1):38–53
4. Oden C, Ercil A, Buke B (2003) Combining implicit polynomials and geometric features for hand recognition. Pattern Recognit Lett 24(13):2145–2152
5. Subrahmonia J, Cooper DB, Keren D (1996) Practical reliable bayesian recognition of 2D and 3D objects using implicit polynomials and algebraic invariants. IEEE Trans Pattern Anal Mach Intell 18(5):505–519
6. Tarel J, Cooper DB (2000) The complex representation of algebraic curves and its simple exploitation for pose estimation and invariant recognition. IEEE Trans Pattern Anal Mach Intell 22(7):663–674
7. Tasdizen T, Tarel J-P, Cooper DB (2000) Improving the stability of algebraic curves for applications. IEEE Trans Imag Proc 9(3):405–416
8. Taubin G (1991) Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Trans Pattern Anal Mach Intell 13(11):1115–1138
9. Taubin G, Cooper DB (1992) Symbolic and numerical computation for artificial intelligence, chapter 6, Computational mathematics and applications. Academic, London
10. Unel M, Wolovich WA (2000) On the construction of complete sets of geometric invariants for algebraic curves. Adv Appl Math 24:65–87
11. Wolovich WA, Unel M (1998) The determination of implicit polynomial canonical curves. IEEE Trans Pattern Anal Mach Intell 20(10):1080–1090
12. Zheng B, Ishikawa R, Oishi T, Takamatsu J, Ikeuchi K (2009) A fast registration method using IP and its application to ultrasound image registration. IPSJ Trans Comput Vis Appl 1:209–219
13. Zheng B, Takamatsu J, Ikeuchi K (2010) An adaptive and stable method for fitting implicit polynomial curves and surfaces. IEEE Trans Pattern Anal Mach Intell 32(3):561–568

Algebraic Surface

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

Synonyms

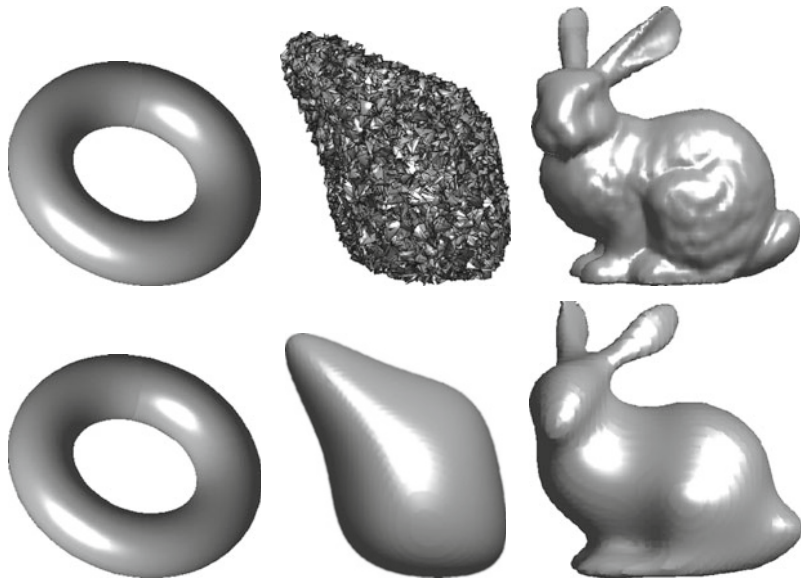
Implicit polynomial surface

Related Concepts

▸Algebraic Curve

Definition

Similar to an algebraic curve, an *algebraic surface* is determined by a 3-D *implicit polynomial* (IP) of degree n:



Algebraic Surface, Fig. 1 Examples of algebraic surfaces. *Top row*: original 3-D data sets of torus, simple shape with noise, and bunny; *bottom row*: resulting algebraic surface fits of degree 4, 4, and 8, respectively

$$
\begin{aligned}
f_n(\mathbf{x}) &= \sum_{0\le i,j,k;i+j+k\le n} a_{ijk}x^i y^j z^k \\
&= (\underbrace{1\ x\ y\ z \ldots\ z^n}_{\mathbf{m}^{\mathrm{T}}})(\underbrace{a_{000}\ a_{100}\ a_{010}\ a_{001}\ \ldots\ a_{00n}}_{\mathbf{a}})^{\mathrm{T}} \\
&= 0, \qquad (1)
\end{aligned}
$$

where $\mathbf{x} = (x, y, z)$ is a 3-D point on a surface, that is, the surface is always represented by f_n's zero level set: $\{\mathbf{x}|\, f_n(\mathbf{x}) = 0\}$. The polynomial function can be denoted by an inner product of two vectors: monomial vector $\mathbf{m}$ and coefficient vector $\mathbf{a}$. For the entries in these vectors, indices $\{i, j, k\}$ can be arranged in different orders, such as *lexicographical order* or *inverse lexicographical order*. In addition, the *homogeneous binary polynomial* of degree r in x, y, and z, $\sum_{i+j+k=r} a_{ijk}x^i y^j z^k$, is called the *r-th degree form* of the IP. The n-th form is also called *leading form*. The degree of algebraic surface is the degree of polynomial: n. An algebraic surface of degree 2 is called a quadric surface, degree 3 a cubic surface, degree 4 a quartic surface, and so on.

Background

In computer vision, representing 3-D surface data sets with algebraic surfaces has been also well studied. It is attractive for vision applications such as 3-D object recognition, pose estimation, and registration. In contrast to other surfaces represented by the functions such as Splines, Fourier, Rational Gaussian, and radial basis function, algebraic surface is superior in such areas as fitting efficiency, few parameters, convenience for calculating algebraic/geometric invariants, and robustness against noise and occlusion. Algebraic surface is also capable of modeling nonstar shapes, open curves, curves that contain gaps, and unordered curve data. However, algebraic surface representation still suffers from some major issues such as the lack of accuracy and stability when representing a complex 3-D shape. Figure 1 shows some examples of algebraic surfaces representing for 3-D surface data sets.

Application and Theory

Algebraic surface representation is mainly attractive for vision applications such as 3-D object registration or pose estimation [2, 4, 8, 10, 11, 13] and recognition [3, 6, 8, 11]. To achieve those purposes, many efforts have been made in three topics: surface fitting, algebraic/geometric invariants, and 3-D object registration. The first topic faces the problem of how to fit an algebraic surface to a complex 3-D shape accurately and stably; the second topic focuses on the problem of how to extract algebraic or geometric invariants from a 3-D shape-representing polynomial; and the third topic concentrates on the task of how to estimate

the rigid transformation relationship between two algebraic surfaces representing the same object in different positions.

Surface Fitting

There have been great improvements concerning algebraic surface fitting with its increased use during the late 1980s and early 1990s [10]. Recently, new robust and consistent fitting methods such as 3L fitting [1], gradient-one fitting with Rigid regression [5, 9], and degree-adaptive fitting [14] have been proposed to make the algebraic surface representation more feasible for vision applications.

Algebraic/Geometric Invariants

The main advantage of algebraic surfaces for recognition is the existence of algebraic/geometric invariants, which are functions of the polynomial coefficients that do not change after a coordinate transformation. The algebraic/geometric invariants that are found by Taubin and Cooper [11], Teral and Cooper [8], and Keren [3] are global invariants and are expressed as simple explicit functions of the coefficients. Another set of invariants that have been mentioned by Wolovich et al. is derived from the covariant conic decompositions of implicit polynomials [12].

3-D Object Registration

In prior literatures such as [7, 11], the global shape registration is performed through single (non-iterative) computation after obtaining the central and oriented information extracted from polynomial coefficients. An iterative method in [13] is proposed by using the distance metric generated from polynomial gradient field and fast polynomial coefficient transformation.

References

1. Blane M, Lei ZB, Cooper DB (2000) The 3L algorithm for fitting implicit polynomial curves and surfaces to data. IEEE Trans Pattern Anal Mach Intell 22(3):298–313
2. Forsyth D, Mundy JL, Zisserman A, Coelho C, Heller A, Rothwell C (1991) Invariant descriptors for 3D object recognition and pose. IEEE Trans Pattern Anal Mach Intell 13(10):971–992
3. Keren D (1994) Using symbolic computation to find algebraic invariants. IEEE Trans Pattern Anal Mach Intell 16(11):1143–1149
4. Keren D, Cooper D, Subrahmonia J (1994) Describing complicated objects by implicit polynomials. IEEE Trans Pattern Anal Mach Intell 16(1):38–53
5. Sahin T, Unel M (2005) Fitting globally stabilized algebraic surfaces to range data. Proc IEEE Conf Int Conf Comp Visi 2:1083–1088
6. Subrahmonia J, Cooper DB, Keren D (1996) Practical reliable bayesian recognition of 2D and 3D objects using implicit polynomials and algebraic invariants. IEEE Trans Pattern Anal Mach Intell 18(5):505–519
7. Tarel J, Cooper DB (2000) The complex representation of algebraic curves and its simple exploitation for pose estimation and invariant recognition. IEEE Trans Pattern Anal Mach Intell 22(7):663–674
8. Tarel J-P, Civi H, Cooper DB (1998) Pose estimation of free-form 3D objects without point matching using algebraic surface models. In: Proceedings of IEEE Workshop Model Based 3D Image Analysis, Mumbai, pp 13–21
9. Tasdizen T, Tarel J-P, Cooper DB (2000) Improving the stability of algebraic curves for applications. IEEE Trans Imag Process 9(3):405–416
10. Taubin G (1991) Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. IEEE Trans Pattern Anal Mach Intell 13(11):1115–1138
11. Taubin G, Cooper DB (1992) Symbolic and numerical computation for artificial intelligence, chapter 6. In: Donald BR, Kapur D, Mundy JL (eds) Computational Mathematics and Applications. Academic, London
12. Wolovich WA, Unel M (1998) The determination of implicit polynomial canonical curves. IEEE Trans Pattern Anal Mach Intell 20(10):1080–1090
13. Zheng B, Ishikawa R, Oishi T, Takamatsu J, Ikeuchi K (2009) A fast registration method using IP and its application to ultrasound image registration. IPSJ Trans Comput Vision Appl 1:209–219
14. Zheng B, Takamatsu J, Ikeuchi K (2010) An adaptive and stable method for fitting implicit polynomial curves and surfaces. IEEE Trans Pattern Anal Mach Intell 32(3):561–568

Analytic Reflectance Functions

▸Reflectance Models

Animat Vision

Dana H. Ballard
Department of Computer Sciences, University of Texas at Austin, Austin, TX, USA

Synonyms

Active vision; Purposive vision

Definition

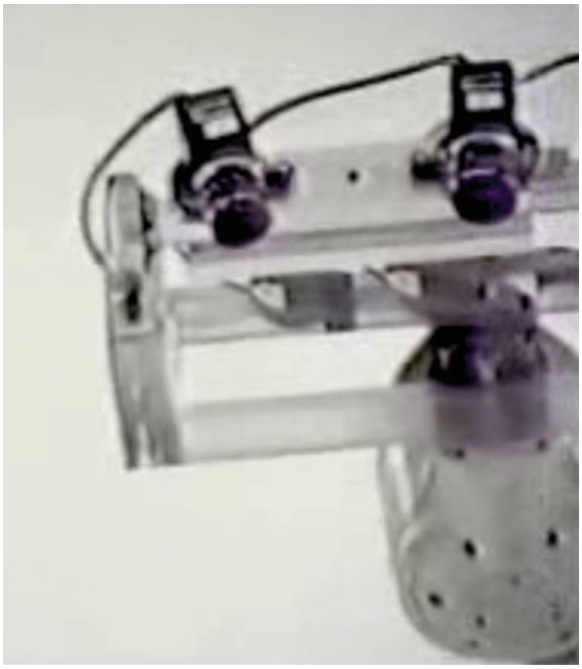
Animat vision is the computational study of the visual systems used by animals, with special attention to the binocular systems used by humans. For human vision, the goal is to show how the characteristics of the human eye movement system can be used to make the computation of needed information more efficient.

Background

The field of computer vision was given an enormous impetus by the publication of Rosenfeld's seminal book *Picture Processing by Computer* [28] in 1969, but in the 1970s the research focus shifted to human vision with the exciting new formulations of *early vision* that recognized that the human visual system devoted enormous resources to extracting cues such as binocular disparity, color, and motion from their composite representation in the initial image. Two groups were especially influential: the group at MIT headed by David Marr and Tomas Poggio [21] and the group at SRI headed by Harry Barrow and Martin Tanenbaum [8]. David Marr in particular had an enormous effect on the field and his book, *Vision*, is a classic [20].

While the early vision paradigm had a wonderful impact of defining computation in vision, by the early 1980s it was apparent that the computations defined on static images were mathematically delicate and could only be tamed with exceptional ingenuity. Thus the idea evolved that perhaps a moving camera, with known movement parameters, would help. An early effort was undertaken at MIT, but the first complete working system was built at the University of Pennsylvania by Ruzena Bajcsy who coined the term *active perception* to describe it [5]. That system was unveiled at a computer vision conference in northern Michigan run by Avi Kak and had an instantaneous acceptance among the researchers present.

Very shortly afterwards, Christopher Brown and the author built a similar system that had a significant advantage. Brown was tracking video processing pipeline computers and realized that this evolving computer architecture, when combined with a servo-driven binocular camera system, would allow the new computations to be realized in real time. The complete system is shown in Fig. 1. Subsequently the appearance of video-rate graphics cards capable of doing real-time image operations served to spur progress. Originally driven by the needs of the computer games industry, researchers quickly realized that now a large amount of the expensive visual calculations could be done in real time. The net result is that animat/active vision moved to lower-cost mobile robotic platforms with the result that robots using a moving cameras on mobile platforms are now commonplace.



Animat Vision, Fig. 1 The University of Rochester real-time servo-driven robotic "head" mounted on its large PUMA "body." Similar systems were built at KTH in Stockholm, Carnegie Mellon University, MIT, the University of Pennsylvania, as well as many other places

Along the mainstream path of robotic animats, Rodney Brooks at MIT, perhaps inspired by Shimon Ullman's concept of *visual routines*, realized that the jobs that vision had to do now became preeminent and built MIT's humanoid robot *Cog* to focus on task-based vision. The particular architecture advocated may not have caught on, but the point was made and the system has been enormously influential. Two

diverse communities – robotics and psychology – have been working on cognitive architectures for managing complex tasks that take a more integrated approach to vision and action, and both have recognized that the ultimate model architecture will have a hierarchical structure, e.g., [3, 11, 12, 16, 23]. Robotics researchers have gravitated to a three-tiered structure that models strategic, tactical, and detailed levels in complex behavior [10].

Theory

Animat vision, like its larger cousin active vision, is a paradigm with a huge number of important papers outstanding, with the consequence that it is only possible to provide the barest of outlines here. The interested reader is referred to some of the early papers [2, 6, 32]. Here we will demonstrate the impact on the calculation of early vision and introduce some more recent developments.

Consequences for Early Vision

Consider the problem of computing just one of the useful early vision representations, that of *optic flow*. Three-dimensional motion due to a moving observer induces the projection of two-dimensional motion on the retina. If the time-varying image function $f(x, y, t)$ represents only this effect, then the differential equations that represent the relationship between optic flows $(u(x, y, t), v(x, y, t))$ can be related to changes in photometric intensities $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial t}$ with the equation, captured at the sensor to the image intensity M by

$$\frac{\partial f}{\partial x}u + \frac{\partial f}{\partial y}v + \frac{\partial f}{\partial t} = 0 \qquad (1)$$

The conundrum of early vision can be easily apparent: At each point in the image x, y at time t, there is a single equation in two unknowns u, v. Poggio famously characterized this as an "ill-posed problem." A plethora of solution methods were tried, but they almost all involved integrating information across the optic array, a very delicate process. In contrast, moving the camera view point immediately solves this problem. If the motion is under the control of the human observer, then, say for the case of horizontal motion, to a first approximation one can assume a reliable estimate for u, and of course the system reduces to a well-behaved two equations in two unknowns. This kind of simplification surfaces again and again in early vision, and many novel instances of this kind of constraint still remain to be discovered.

The Importance of Eye Fixations

Yarbus's original work in gaze recordings [37] in the 1950s and 1960s revealed the enormous importance of gaze in revealing the underlying structure of human cognition. From this perspective, it is somewhat surprising that the first significant computational theory of vision [20] postponed the study of gaze as well as any influence of cognition of the extraction of information from the retinal array. In his "principle of least commitment," Marr argued the case for the role of the cortex in building elaborate goal-independent descriptions of the physical world. Perhaps as a consequence, when researchers took on the task of defining the mechanisms for directing gaze deployment, these turned out to be predominantly image based [18, 19, 36]. These theories have been compelling, but have many drawbacks. They usually cannot predict the exact landing points and typically leave more than 30 % of the fixations unaccounted for.

Recent experiments show that fixations are extracting very specific information needed by the human subject's ongoing *task* [14, 34]. The task context introduces enormous economies into this process that are very obvious: If a subject needs to pick up a red object, the search for that object can be limited to just red portions of the image; vast amounts of extraneous detail can be neglected. The visual information-gathering specificity of almost every portion of every task will introduce similar economies. Knowledge of task also has the promise of interpreting a substantial literature devoted to "change blindness." Subjects fail to notice large changes between successive images or movie frames. While the exact reason for this has been the subject of controversy [22], the problem may be resolvable if one has access to the viewer's cognitive agenda. On agenda changes are noticed and off agenda changes are not.

Theoretical Breakthroughs in Task Modeling

What experiments testing the information extracted during a fixation have lacked is a theory that accounts for the role of the cognitive processes that are controlling the subject's behaviors. What form should such a theory take? There have been several enormous theoretical advances, mostly from the fast emerging field of machine learning, that promise to have an enormous impact on quantitatively testable theories of cognition. The requirements of animat vision suggest that such a cognitive theory will have three important elements: (1) probabilistic representations, (2) the use of reward in learning, and (3) embodied cognitive architectures.

1. *Probabilistic methods.* There is rapidly increasing recognition that the brain is a probabilistic device and maintains a variety of mechanisms for calculating the statistical model of the world around it and its actions upon that world. To handle this a major new representational formalism has been developed that goes under the name *graphical models*. Originally developed by Pearl [26], such models have seen refinement as a general way to factor complex statistical interdependencies into to locally calculable quantities. The result is that the maintenance of elaborate statistical dependencies has become practical. Furthermore, Bayesian models of these interdependencies have proven their worth in characterizing many different observations in visual perceptions [35, 38].
2. *Reinforcement Learning.* A second breakthrough has been the development of model-making algorithms that are programmed by reward. It has long been appreciated that the brain must have mechanisms to learn complex behaviors and that these mechanisms must be steered by some scalar affinity signal. For the dominant effects the neurotransmitter dopamine has been implicated as the major signaling mechanism. Schultz, Glimcher, and others have made the connection between dopamine and reinforcement learning algorithms [17, 27, 33], the latter which by themselves have seen rapid development [1, 9]. Reinforcement learning algorithms are in their infancy but hold the promise of being and integral part of a comprehensive theory of animat vision learning.
3. *Embodied Cognition.* As emphasized by a number of researchers, the brain cannot be understood in isolation as so much of its structure is dictated by the body it finds itself in and the world that the body has to survive in [7, 13, 24, 25, 29]. This has special important implications, particularly for the cognitive architectures, because the brain can be dramatically simpler that it could ever be without these encasing milieus. The reason is that the brain does not have to replicate the natural structure of the world or the special ways of interacting with it taken by the body but instead can have an internal structure that implicitly and explicitly anticipates these commitments. The brain just has to have an interface that allows successful interactions with the world, but does not have to explicitly model all the detailed consequences of the actions taken. This realization opens up a way to address the challenge of making the leap from the apparent simplicity of the observed behaviors to the complexity of the brain-body-world system that produces them and that is to see the behaving body itself as a laboratory instrument. From this vantage point, the momentary disposition of the eyes, head, and hands during the course of behavior reveals essential details about the underlying cognitive program that is making those choices.

Open problems

Although the importance of body in cognition has been stressed at least since Merleau-Ponty, until the middle of 1980s, it was only practical to study very controlled circumstances such as those made by an experimental subject seated in front of a small display.

The research program at Rochester pioneered the study of embodied, visually driven behaviors by the development of innovative laboratory equipment and techniques. With Pelz at the Rochester Institute of Technology [4], they were the first laboratory that were able to track the eyes inside a head-mounted display. This capability allowed the exploitation of another recent development: Virtual Reality(VR). It is now straightforward to render scenes in real time from a moving observer's vantage point that are extraordinarily close to the real thing. Thus a person can have the compelling illusion of being in a fictional world that at the same time is under experimental control. This capability, in turn, has allowed researchers to address many new experimental questions for the first time. For example, one can study a person's disposition

of visual resources in these virtual worlds by using the eye trackers inside the head-mounted display to manipulate the information that is available at each fixation [14, 30, 34].

Now flexible portable instrumentation can be attached to the body during the course of extended natural behaviors. Eye tracking capability that started out requiring subjects to be restrained in a bite bar has evolved to the point where portable trackers can be worn during a squash match. Head, hand, and body movements, even those of the facial muscles during expressions, can be reliably captured at high data rates during tea making, athletics, and everyday conversation. The new instrumentation opens up the possibility of acquiring large amounts of such data at millisecond time scales during these natural behaviors and thus provides access to the essential choices made in directing behavior under natural circumstances.

Obtaining such data from behavior and modeling it has led to another new question: How does one become confident that the models one builds are accurate? Answering this question has led to another new development and that is simulated human modeling. It is now possible to create models of humans that have the degrees of freedom of the skeletal system and also the capabilities of the binocular vision system [15, 31]. Thus one can build a human avatar that acts out the cognitive models obtained by fitting human data. A bonus is that one can test the models in completely new situations that were not part of the original human data gathering and observe their performance. This in turn can lead to an iterative refinement of the models and new experiments. However the most important aspect of this animat vision research avenue is the testing of the embodied cognition hypothesis with a suitably rich model. Our everyday experience and introspection as to the nature of the execution of everyday tasks has proven very misleading as to the brain's underlying representations owing to the artfulness of conscious experience.

References

1. Abeel P, Quigley M, Ng AY (2006) Using inaccurate models in reinforcement learning. In: International conference on machine learning, Pittsburgh
2. Aloimonos J, Bandopadhay A, Weiss I (1988) Active vision. Int J Comput Vis 1(4):333–356
3. Arkin R (1998) Behavior based robotics. MIT, Cambridge
4. Babcock JS, Pelz JB, Peak J (2003) The wearable eye-tracker: a tool for the study of high-level visual tasks. In: Proceedings of the MSS-CCD2003
5. Bajcsy R (1988) Active perception. Proc IEEE 76:966–1005
6. Ballard DH (1991) Animate vision. Artif Intell 48(1): 57–86
7. Ballard D, Hayhoe M, Pook P (1997) Deictic codes for the embodiment of cognition. Behav Brain Sci 20:723–767
8. Barrow HG, Tanenbaum JM (1978) Computer vision systems. Academic, New York, pp 3–26
9. Barto AG, Mahadevan S (2004) Recent advances in hierarchical reinforcement learning. Discret Event Dyn Syst 13:341–379
10. Bonasso RP, Firby RJ, Gat E, Kortenkamp D, Miller DP, Slack MG (1997) Experiences with an architecture for intelligent reactive agents. J Exp Theor Artif Intell 9:237–256
11. Brooks RA (1986) A robust layered control system for a mobile robot. IEEE J Robot Autom RA-2(1):14–23
12. Bryson JJ, Stein LA (2001) Modularity and design in reactive intelligence. In: International joint conference on artificial intelligence, Seattle
13. Clark A (1999) An embodied model of cognitive science? Trends Cogn Sci 3:345–351
14. Droll JA, Hayhoe MM, Triesch J, Sullivan BT (2005) Task demands control acquisition and storage of visual information. J Exp Psychol Hum Percept Perform 31: 1415–1438
15. Faloutsos P, van de Panne M, Terzopoulos D (2001) The virtual stuntman: dynamic characters with a repertoire of motor skills. Comput Graph 25:933–953
16. Firby RJ, Kahn RE, Prokopowicz PN, Swain MJ (1995) An architecture for vision and action. In: Fourteenth international joint conference on artificial intelligence, Montréal, pp 72–79
17. Hayden BY, Platt ML (2007) Temporal discounting predicts risk sensitivity in rhesus macaques. Curr Biol 17:49–53
18. Itti L, Baldi P (2005) A principled approach to detecting surprising events in video. In: IEEE international conference on computer vision and pattern recognition (CVPR), San Diego, vol 1, pp 631–637
19. Koch C, Ullman S (1985) Shifts in selective visual attention: towards the underlying neural circuitry. Hum Neurobiol 4:219–227
20. Marr D (1982) Vision: a computational investigation into the human representation and processing of visual information. Freeman, San Francisco
21. Marr D, Poggio T (1979) A computational theory of human stereo vision. Proc R Soc Lond B 204:301–328
22. Most SB, Scholl BJ, Clifford ER, Simons DJ (2005) What you see is what you set: sustained inattentional blindness and the capture of awareness. Psychol Rev 112:217–242
23. Newell A (1990) Unified theories of cognition. Harvard University Press, Cambridge
24. Noe A (2005) Action in perception. MIT, Cambridge/London
25. O'Regan JK, Noe A (2001) A sensorimotor approach to vision and visual consciousness. Behav Brain Sci 24: 939–973
26. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Mateo

27. Platt ML, Glimcher PW (1999) Neural correlates of decision variables in parietal cortex. Nature 400:233–238
28. Rosenfeld A (1969) Digital picture processing. Academic, Orland
29. Roy DK, Pentland AP (2002) Learning words from sights and sounds: a computational model. Cogn Sci 26:113–146
30. Shinoda H, Hayhoe MM, Shrivastava A (2001) What controls attention in natural environments? Vis Res 41: 3535–3546
31. Sprague N, Ballard D (2003) Multiple-goal reinforcement learning with modular sarsa(0). Technical report 798, Computer Science Department, University of Rochester
32. Terzopoulos D, Rabie TF (1997) Animat vision: active vision in artificial animals. Videre J Comput Vis Res 1(1): 2–19
33. Tobler PN, Fiorillo CD, Schultz W (2005) Adaptive coding of reward value by dopamine neurons. Science 307: 1642–1645
34. Triesch J, Ballard D, Hayhoe M, Sullivan B (2003) What you see is what you need. J Vis 3:86–94
35. Weiss Y, Simoncelli EP, Adelson EH (2002) Motion illusions as optimal percepts. Nat Neurosci 5:598–604
36. Wolfe J (1994) Guided search 2.0. a revised model of visual search. Psychon Bull 1:202–238
37. Yarbus AL (1967) Eye movements and vision. Plenum Press, New York
38. Yuille A, Kersten D (2006) Vision as bayesian inference: analysis by synthesis? Trends Cogn Sci 10:301–308

Anisotropic Diffusion

▸Diffusion Filtering

Aperture Ghosting

▸Lens Flare and Lens Glare

Appearance Scanning

▸Recovery of Reflectance Properties

Appearance-Based Human Detection

William Robson Schwartz
Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte, MG, Brazil

Synonyms

Appearance-based pedestrian detection

Related Concepts

▸Object Detection

Definition

Human detection may be seen as a classification problem with two classes: human and nonhumans, in which the latter class is composed of background samples containing anything but humans. When the appearance-based human detection is employed, a large number of examples of human and nonhumans are considered to capture different poses, backgrounds, and occlusion situations through the extraction of feature descriptors so that a machine learning method can be used to classify samples as belonging to either one of the classes.

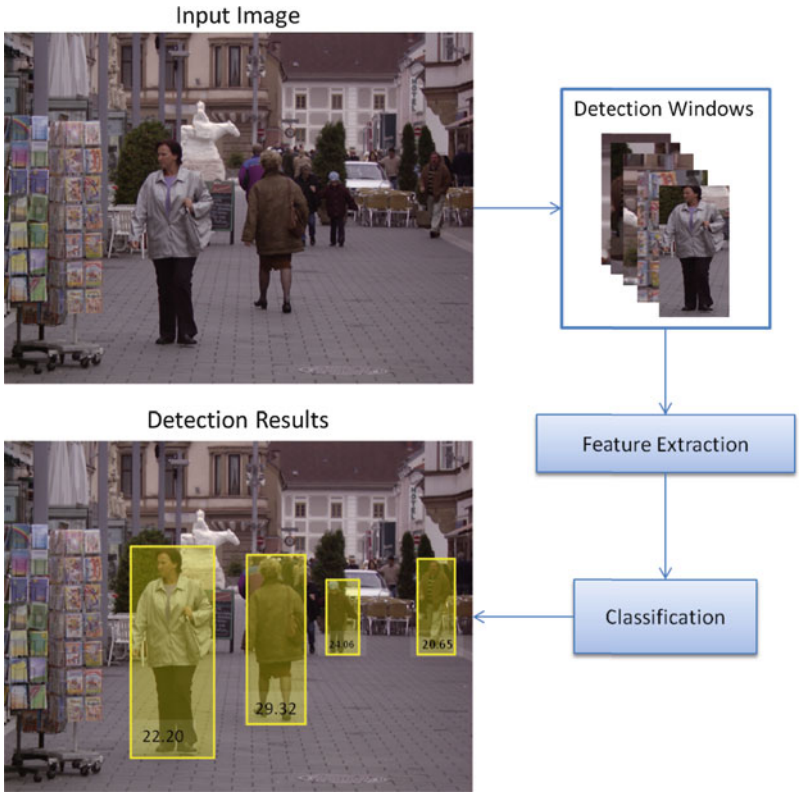
Background

Due to the large number of applications that require information regarding people's location, such as autonomous vehicles, surveillance, and robotics, finding people in images or videos presents large interest of the community. Even though widely studied in recent years [1], the human detection problem is still a challenge due to the wide variety of poses, clothing, background, and partial occlusions, which generate a large number of person's appearances.

Two main approaches have been explored in the human detection literature. The first class of methods consists of a generative process that combines detected parts of the human body according to a model. The second class considers statistical analysis through the use of machine learning techniques to classify a feature vector composed of low-level feature descriptors extracted from a detection window. This approach, also referred to as appearance-based, captures the appearance information and focuses on the discrimination between human and nonhuman samples.

Theory

Appearance-based human detection presents two important aspects: feature extraction and classification. Once both aspects have been considered, the training and test (detection) steps can be performed.



Appearance-Based Human Detection, Fig. 1 Example of the human detection process. Image sample extracted from the INRIA Person dataset [2]

The feature extraction is responsible for capturing the visual information from the scene, such as the presence of strong vertical edges, homogeneous textured clothing, or color constancy in the face. Such characteristics, useful for human detection, will be extracted using low-level feature descriptors. It has been shown that the combination of these characteristics improves detection results [3]. Among the most used feature descriptors are the histograms of oriented gradients (HOG) [2], local binary patterns (LBP) [4], and Haar wavelet-based features [5].

The second relevant aspect is the choice of a machine learning method capable of classifying between humans and non-humans by giving higher importance to those descriptors that best distinguish between the two classes. Among the most employed methods are the linear discriminant analysis (LDA), neural networks (NN), support vector machines (SVM), and partial least squares (PLS).

The training step is responsible for learning parameters of the machine learning methods such that the differences between the two classes can be properly captured. For that, features are extracted from multiple samples from both classes, and the descriptors are stored in feature vectors. It is important to emphasize that a good training set is important to assure that variations of the human appearances are captured. Each classification method presents a different way of learning the differences. For example, while SVM finds support vectors that maximize the margins between both classes, PLS will give more weight to those dimensions of the feature vector that best discriminate between the classes. In addition, it is important to note that a good training set is also important to assure that variations of the human appearances are captured.

Once the training has been performed, a sliding window is passed in the image at multiple scales to locate humans at different locations and scales. For each location, features are extracted and stored in a feature vector, which is then presented to the classifier. The output for each detection window is a value that reflects the probability or confidence in which a human is located inside the detection window. Figure 1 illustrates the detection process of a typical appearance-based human detection method.

Application

In general, the human detection is of interest in any application that falls inside the *Looking at People* [6] (domain which focuses primarily in analyzing images and videos containing humans). For example, a human detector can be used to provide the location of each agent in a scene so that tasks such as tracking, re-identification, action, and activity recognition can be executed by a surveillance system. In addition, a human detector can be executed in the domain on autonomous navigation, where the location of the pedestrians will be used as information for path planning. Furthermore, the use of human detection systems embedded in vehicles may be very useful to assure pedestrian safety [7].

References

1. Enzweiler M, Gavrila DM (2009) Monocular pedestrian detection: survey and experiments. IEEE Trans Pattern Anal Mach Intell 31(12):2179–2195
2. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE conference Computer Vision and Pattern Recognition (CVPR), San Diego, pp 886–893
3. Schwartz WR, Kembhavi A, Harwood D, Davis LS (2009) Human detection using partial least squares analysis. In: IEEE International Conference on Computer Vision, Kyoto, pp 24–31
4. Wang X, Han TX, Yan S (2009) An HOG-LBP human detector with partial occlusion handling. In: IEEE International Conference on Computer Vision, Kyoto, pp 32–39
5. Viola P, Jones MJ, Snow D (2005) Detecting pedestrians using patterns of motion and appearance. Int J Comput Vis 63(2):153–161
6. Gavrila DM (1999) The visual analysis of human movement: a survey. Comput Vis Image Underst 73(1):82–98
7. Gandhi T, Trivedi M (2007) Pedestrian protection systems: issues, survey, and challenges. IEEE Trans Intell Transp Syst 8(3):413–430

Appearance-Based Human Tracking

Bohyung Han
Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH), Pohang, South Korea

Synonyms

Human appearance modeling and tracking

Related Concepts

▸Human Pose Estimation

Definition

Appearance-based human tracking is a human tracking algorithm, where the measurement is based on the appearance of human such as color, texture, shape, and their combination.

Background

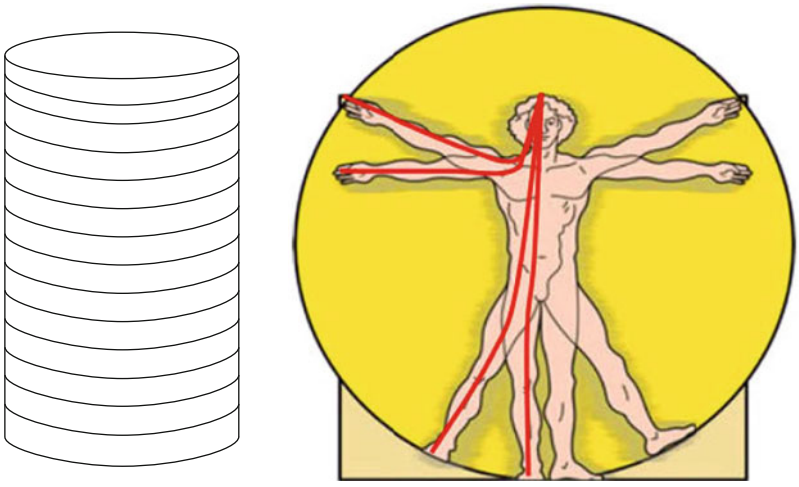
Various human tracking algorithms have been proposed so far, but the focus of each algorithm is different. Appearance-based human tracking is an algorithm to track human based on the similarity between the existing appearance model and the observation in image; the control and search algorithm in tracking can be arbitrary. Several different features have been used for human tracking including color, edge, gradient, texture, and shape, and multiple features are often integrated together for more robust observations. The appearance of human based on the features is represented by density function, histogram, template, and other descriptors. For the representation of human body, the spatial layout of the features is typically employed to model the appearance of human. The appearance models can be constructed for the entire human body, a few subregions of the human body, or the individual articulated human body parts separately, depending on the target state spaces and tracking algorithms.

Generic tracking algorithms can be employed to track a blob-based human with a simple appearance model in a low-dimensional state space. However, in case of articulated human body tracking, the dimensionality of target state space is typically very high, and the inference procedure of the complex human body configuration is generally complex; more efficient and specialized tracking algorithms are required.

Theory

Methodologies of Appearance Models

Two aspects – integrated features and mathematical representation techniques – are considered to characterize the appearance-based human tracking.



Appearance-Based Human Tracking, Fig. 1 Some examples to model spatial layout of a person in human appearance modeling. (*Left*) The appearance model is developed for each height slice of a person. (Fig. 2 in [1]) (*Right*) The path length is the distance from the *top* of the head to a given point on the path. (Fig. 2 in [9])

Features Integrated

The appearance for human tracking is modeled with color [1–3], silhouette [4, 5], shape [6, 7], edge [4, 8], or texture.

Representation Method

Some appearance modeling techniques assume that the appearance of human body is consistent horizontally. With the assumption, human body is represented with multiple histograms based on a cylinder model as in the left subfigure in Fig. 1 [1] or density function [3]. Another method is a path-length model [9], where the spatial variations are modeled by the distance from head along the shape of the person as shown in the right subfigure in Fig. 1 and spatial-feature distribution is constructed for appearance modeling. Template is also frequently used [7, 10], and probabilistic template is integrated in [11].

Acquisition and Maintenance of Appearance Models

The appearances may be fixed throughout the sequence or adaptive to the variations of human body appearance. The initialization of the appearance can be performed based on (manual or automatic) human detection. In [12], the appearance of each body part of human is learned in an online manner based on simple features obtained in off-line process.

Tracking Control and Search

Human tracking can be classified into two types based on the description method of human body; one is blob-based tracking, and the other is articulated human body tracking. In case of blob-based tracking, tracking algorithm is simple, and the state space of the target is typically low dimensional. The algorithms in this type have no big difference from generic tracking algorithms for other objects; a major difference is that human tracking algorithms often divide target into a few sub-regions based on appearance consistency to improve measurement accuracy. However, the articulated human body tracking involves very high-dimensional state space (typically more than 20 dimension) and complicated probabilistic inference procedures; efficient tracking control and search algorithms are required to handle the challenges such as annealing [4, 11] and covariance sampling [5] in particle filter framework.

Application

Appearance-based human tracking has a lot of potential applications such as event detection and video understanding, pedestrian detection and tracking for intelligent vehicles and vision-based user interface in computer games.

References

1. Mittal A, Davis L (2003) M2tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. Int J Comput Vis 51(3):189–203
2. Ramanan D, Forsyth D, Zisserman A (2005) Strike a pose: tracking people by finding stylized poses. In: IEEE conference on computer vision and pattern recognition (CVPR), vol 1. IEEE Computer Society, Los Alamitos, pp 271–278
3. Kim K, Davis LS (2006) Multi-camera tracking and segmentation of occluded people on ground plane using search-guided particle filtering. In: European conference on computer vision (ECCV), Graz, vol 3, pp 98–109
4. Deutscher J, Reid I (2005) Articulated body motion capture by stochastic search. Int J Comput Vis 61:185–205
5. Sminchisescu C, Triggs B (2003) Estimating articulated human motion with covariance scaled sampling. Int J Robot Res 22(6):371–392
6. Haritaoglu I, Harwood D, David LS (2000) W4: real-time surveillance of people and their activities. IEEE Trans Pattern Anal Mach Intell 22:809–830
7. Lim H, Camps O, Sznaier M, Morariu V (2006) Dynamic appearance modeling for human tracking. In: IEEE conference on computer vision and pattern recognition (CVPR), New York, vol 1, pp 751–757
8. Poon E, Fleet D (2002) Hybrid Monte Carlo filtering: edge-based people tracking. In: IEEE workshop on motion and video computing, Orlando, pp 151–158
9. Yoon K, Harwood D, Davis LS (2006) Appearance-based person recognition using color/path-length profile. J Vis Commun Image Represent 17(3):605–622
10. Cham TJ, Rehg J (1999) A multiple hypothesis approach to figure tracking. In: IEEE conference on computer vision and pattern recognition (CVPR), Fort Collins, vol 1, pp 239–245
11. Balan A, Black M (2006) An adaptive appearance model approach for model-based articulated object tracking. In: IEEE conference on computer vision and pattern recognition (CVPR), New York, vol 1, pp 758–765
12. Ramanan D, Forsyth D, Zisserman A (2007) Tracking people by learning their appearance. IEEE Trans Pattern Anal Mach Intell 29(1):65–81

Appearance-Based Pedestrian Detection

►Appearance-Based Human Detection

Articulated Pose Estimation

►Human Pose Estimation

Asperity Scattering

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

Synonyms

►Surface scattering; ►Velvety reflectance

Related Concepts

►Retroreflection; ►Lambertian Reflectance; ►Surface Roughness

Definition

The "asperities" can be of various nature, hairtips, dust, "fluff", local high curvature spots or ridges (the term derives from scattering by powdered materials where the "asperities" are sharp edges like on broken glass).

Background

The reflectance of natural, opaque, and rough surfaces [1, 2] can be described by the Bidirectional Reflectance Distribution Function (BRDF) [6]. BRDFs that are common and well-known are those of Lambertian, perfectly diffusely scattering surfaces and of specular surfaces. Such surfaces scatter light in all directions (diffuse scattering) or primarily in the mirror direction (specular reflection). However, natural surfaces can scatter light in many other ways. Asperity scattering adds a "surface lobe" to the usual diffuse, backscatter, and specular lobes of rough surfaces. It is an important effect in many materials that are covered with a thin layer of sparse scatterers such as dust or hairs. In the common case that single scattering predominates, asperity scattering adds important contributions to the structure of the occluding contour and the edge

of the body shadow. This is the case because the BRDF is inversely proportional to the cosines of both illumination and viewing angles. The BRDF is generally low (and typically negligible), except when either the illuminating rays or visual directions graze the surface.

Because asperity scattering selectively influences the edges in the image of an object, it has a disproportionally (as judged by photometric magnitudes) large effect on (human) visual appreciation. It is a neglected but often decisive visual cue in the rendering of human skin. Its effect is to make smooth cheeks to look "velvety" or "peachy" (the appearances of both velvet and "peachy" skin are dominated by asperity scattering), that is to say, soft. This is a most important aesthetic and emotional factor that is lacking from Lambertian (looks merely dullish, paperlike), "skin type" BRDF (looks like glossy plastic), or even translucent (looks "hard", vitreous) types of rendering.

Theory

Asperity scattering is due to scattering by a sparse "cloud cover" of the surface with essentially point scatterers. In sparse distributions of scatterers, one may assume that single scattering predominates. Then parameters of interest are the geometry of the cloud and the nature of the single scatterers. For this case, a physical, geometrical optical model was derived [3] and experimental data gathered [5].

It is also possible to fit asperity scattering characteristics in a convenient, simplified formula (note that basic physical constraints should hold, e.g., non-negativity, energy conservation, and Helmholtz reciprocity) [4]. For instance, for a surface element with unit (outward) normal n , irradiated from the direction (unit vector) i and viewed from the direction (unit vector) j, the following BRDF model

$$V(\mathbf{i},\mathbf{j},\mathbf{n},a) = \frac{1}{\pi}\frac{a}{a+(\mathbf{i}\cdot\mathbf{n})(\mathbf{j}\cdot\mathbf{n})}, \qquad (1)$$

describes a "surface lobe" such as one observes in black velvet cloth or peach skin. The parameter a determines the width of the lobe. (A similar behavior results if one substitutes $(\mathbf{i}\cdot\mathbf{n})+(\mathbf{j}\cdot\mathbf{n})$ for $(\mathbf{i}\cdot\mathbf{n})(\mathbf{j}\cdot\mathbf{n})$.) The albedo is found to be

$$A_V(\mathbf{i},\mathbf{n},a) = \frac{2a}{(\mathbf{i}\cdot\mathbf{n})^2}\left(\mathbf{i}\cdot\mathbf{n} + a\log\frac{a}{a+\mathbf{i}\cdot\mathbf{n}}\right), \qquad (2)$$

which has a lowest value

$$2a\left(1+\log\left(\frac{a}{1+a}\right)^a\right) \approx 2a + 2\log a a^2 + \ldots \qquad (3)$$

at normal incidence and rises monotonically to unity at grazing incidence. (For black velvet $a \ll 1$.) Other possibilities for simplified formulations may be found in graphics as so-called velvet shaders. However, care should be taken that many of these rendering applications do not fulfill the above mentioned basic physical constraints.

Open Problems

BRDFs of natural surfaces can probably be categorized into about a dozen different modes. Currently, only the forward, backward, diffuse, and surface scattering modes have been described by formal optical models.

Reflectance estimation from images suffers from image ambiguities. Prior knowledge on the reflectance statistics of natural materials plus formal descriptive models for the common modes of natural BRDFs can constrain this problem.

References

1. CUReT: Columbia–Utrecht reflectance and texture database. http://www.cs.columbia.edu/CAVE/curet
2. Dana KJ, van Ginneken B, Nayar SK, Koenderink, JJ (1999) Reflectance and texture of real-world surfaces, ACM Trans on Graphics, 18(1):1–34
3. Koenderink JJ, Pont SC (2003) The secret of velvety skin. Mach Vis Appl 14:260–268
4. Koenderink JJ, Pont SC (2008) Material properties for surface rendering. Int J Comput Vis Biomech 1(1):43–53

5. Lu R, Koenderink JJ, Kappers AML (1998) bidirectional reflection distribution functions) of velvet. Appl Opt 37(25):5974–5984
6. Nicodemus FE, Richmond JC, Hsia JJ (1977) Geometrical considerations and nomenclature for reflectance. National bureau of standard US monograph, 160

Automatic Gait Recognition

►Gait Recognition

Automatic Scale Selection

►Scale Selection

Automatic White Balance (AWB)

►White Balance