Optic-Flow Based Motion Analysis for Animation

Chunrong Yuan[1], Irene Cheng[2], Anup Basu[2] and Hanspeter A. Mallot[1]

[1] Chair for Cognitive Neuroscience, University of Tübingen, Germany
[2] Department of Computing Science, University of Alberta, Canada

***Abstract*—Creating graphics animations is often a tedious process that requires days of content creation to make even a short video. In this work, we investigate the role of optic flow in making it easier to automate certain steps in creating animated clips. More specifically, we look into how optic flow can be used to detect and model natural events, such as, waterfall, rainfall, snowfall and so on. An algorithm that can not only detect 2D flow, but also track feature points reliably over video frames is described; followed by experimental results demonstrating its ability in motion analysis and motion synthesis. An example of the steps involved in creating a short animation is also outlined.**

***Index Terms*—Image-Based Animation, Motion Analysis, Motion Synthesis, Optic Flow**

I. INTRODUCTION

Extensive research is reported in the literature on optic flow analysis. Optic flow is the distribution of apparent motion of brightness patterns in an image. Much of the research on optic flow focuses on developing models and methodologies to recover a 2D motion field. While most of the approaches apply the general spatio-temporal constraint, they differ in how the two components (u and v) are solved using additional constraints. One classical solution provided by [1] takes a global approach, which uses a smoothness constraint based on second-order derivatives. The flow vectors are then solved using nonlinear optimization methods. The solution proposed by Lucas-Kanade [2] takes a local approach, which assumes equal flow velocity within a small local neighborhood. A closed-form solution to the flow vectors is then achieved, which involves only first-order derivatives. Some variations as well as combinations of the two approaches can be found in [3].

Substantial research has also been carried out on motion or structure analysis based on optic flow. Some work attempts to recover the egomotion (i.e. camera motion) undergone by the viewer for navigating in a static environment [4]. Given a static viewer, it is possible to analyze the 3D structure or motion of the objects viewed [5]. A somewhat more difficult problem is the separation of egomotion and the independent motion of objects when neither the environment nor the viewer is static [6, 7, 8].

In the computer graphics community, research on image and video based animation has focused primarily on face and body movement synthesis [9, 10, 11]. Adding painting-like effects to image and video was considered in [12]. However, the use of optic flow for animating natural events has been limited.

In this paper, we use optic flow to make it easier to automate certain steps in creating animated videos of natural events. Specifically, we apply robust optic flow techniques to detect motion in video sequences and analyze the motion parameters and characteristics of water particles, which provide valuable information for the animation of natural motion, such as rain and snow. In Section 2, we present a robust algorithm for the measurement of 2D velocity fields. In Section 3, we extend the optic flow algorithm for detection and analysis of a waterfall. Section 4 discusses some of the steps taken to create an animated sequence along with illustrative examples. Finally, Section 5 presents the conclusion and future work.

II. ROBUST OPTIC FLOW ALGORITHM

The first step in optic flow analysis is the recovery of the flow velocity of scene points. Here we use a local approach, which is based on the aforementioned Lucas-Kanade algorithm [2]. The original Lucas-Kanade algorithm is quite simple to realize. Suppose the brightness of a point located at p(x, y) in the image taken at time t is f(x, y, t). Let f_x and f_y be the spatial image gradient, and f_t the temporal image derivative, the local smoothness constraint can be expressed as:

$$\varepsilon(\mathbf{v}) = \sum_{(x,y)\in W} (f_x u + f_y v + f_t) = 0 \tag{1}$$

Where vector $\mathbf{v} = [u, v]^T$ is the 2D velocity of the point p and W is a small neighborhood around p. The flow vector $\mathbf{v}$ can be solved as:

$$\mathbf{v} = \mathbf{G}^{-1}\mathbf{b} \tag{2}$$

with

$$\mathbf{G} = \sum_{(x,y)\in W} \begin{bmatrix} f_x^2 & f_x f_y \\ f_x f_y & f_y^2 \end{bmatrix}, \tag{3}$$

$$\mathbf{b} = -\sum_{(x,y)\in W} \begin{bmatrix} f_x f_t \\ f_y f_y \end{bmatrix}. \tag{4}$$

Manuscript Received on 26 February 2010
E-Mail: yuan@computer.org

The above solution requires that $\mathbf{G}$ be invertible, which means that the image should have gradient information in both x and y directions in the neighborhood W centered at point p. For the purpose of better performance, we apply a point selection process before calculating the flow vector $\mathbf{v}$.

As the matrix $\mathbf{G}$ is symmetric, it can be diagonalized using singular value decomposition as:

$$\mathbf{G} = \mathbf{R}^{-1}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}\mathbf{R} \tag{5}$$

Where $\mathbf{R}$ is a rotation matrix. If we want to have gradient information in both directions, we need to choose the point with the following characteristics:

It has large eigenvalues, i.e. λ_1 and λ_2 should be large.

In order to have a well-conditioned matrix $\mathbf{G}$ for good estimates of $\mathbf{v}$, the ratio of λ_1 / λ_2 should not be too large.

One advantage of point selection lies in the fact that it circumvents the aperture problem. Furthermore, the criterion based on eigenvalues makes it possible to have more reliable and accurate flow vectors as well as more robustness against noise. For the purpose of subpixel estimation of flow vectors, we use an iterative algorithm and carry out a pyramidal implementation, applying a coarse to fine approach, as suggested in [2].

In modeling optic flow, one assumes that the brightness of a point does not change between frames. As a consequence, brightness changes may corrupt the velocity estimates. We can imagine that flow vectors for some points with stronger brightness variations between two successive frames will be less accurate. In the following, we refer to such points as outliers. To remove these outliers resulting from the inherent deficiency of the optic flow model, we apply a further constraint called frontoparallel motion constraint.

Suppose we have calculated the 2D velocity of a set of 2D image points $\{\mathbf{p}_k\}$ in frame t. This means that we can find another set of points $\{\mathbf{q}_k\}$ in frame t+1 with $\mathbf{q}_k = \mathbf{p}_k + \mathbf{v}_k$. We parameterize the motion between the two frames using a 2D rotation $\mathbf{R}$, a 2D translation $\mathbf{t}$ and a scale factor s so that $\mathbf{q}_k = s\mathbf{R}\mathbf{p}_k + \mathbf{t}_k$. Using a least-squares estimation method [13], the three parameters $\{s, \mathbf{R}, \mathbf{t}\}$ can be determined uniquely. For each point $\mathbf{p}_k$, we then calculate a distance measure $d_k = \|(\mathbf{p}_k + \mathbf{v}_k) - (s\mathbf{R}\mathbf{p}_k + \mathbf{t}_k)\|^2$. This distance measure tells us whether the 2D velocity of each point agrees with the motion model specified by $\{s, \mathbf{R}, \mathbf{t}\}$. If a point has a large distance, it can hence be regarded as an outlier.

III MOTION DETECTION AND ANALYSIS

We apply the robust optic flow algorithm for the motion analysis of natural events such as that of water particles. To illustrate the process of motion detection and analysis, we use waterfall sequences as an example. Since video sequences are captured with a handheld camera, we not only need to detect the motion field but also need to separate the waterfall motion from that of the camera.

For motion separation, we apply an approach based on statistical pattern recognition. Without loss of generality, we assume that the camera motion and waterfall motion are different. This means that flow vectors belonging to waterfall would have a different distribution from those caused by camera motion. Let Ω_0 be the set of points whose flow vectors are caused by camera motion and Ω_1 be the set of points belonging to waterfall. The classification of the flow points into two classes Ω_0 and Ω_1 can be achieved from the histogram of the 2D motion field. For more details about the histogram based pattern classification method, please refer to Section 2.1 in [14].

After the flow points caused by camera motion have been identified, we calculate the 2D motion parameters $\{s_c, \mathbf{R}_c, \mathbf{t}_c\}$ of the camera egomotion [13], using only the points belonging to Ω_0. With the egomotion parameters calculated, we can recover the true waterfall motion as $\mathbf{v}_t = \mathbf{v} - \mathbf{v}_c$, where $\mathbf{v}_t$ is the rectified true motion of the waterfall, $\mathbf{v}_c$ the flow velocity caused by camera motion being calculated as $\mathbf{v}_c = s_c\mathbf{R}_c\mathbf{p} + \mathbf{t}_c$.

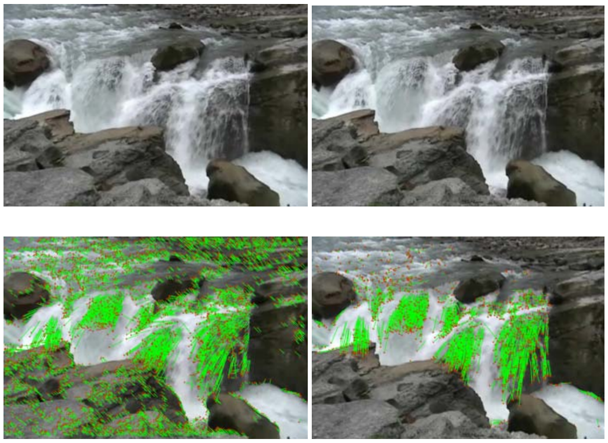

Fig. 1. Detection and rectification of waterfall motion(Color Plate 1)

An example of the separation and rectification of waterfall motion is shown in Fig. 1. On the top row are the two video frames and the bottom left image shows the 2D motion field resulting from our robust optic flow algorithm. One can observe that there exists motion of both camera and waterfall, as the flow vectors in the stone region are caused by camera motion. By using the histogram-based algorithm, we are able to differentiate between the two kinds of motion. With the egomotion parameters calculated, we then eliminate the effect of camera motion and recover the true waterfall motion. Fig. 1 (bottom right) shows the rectified motion field of the waterfall.

Besides waterfall motion detection and separation, we also track individual water points across frames and analyze their motion trajectory. By tracking we mean the localization of each water point in successive video frames based on the calculated flow velocity. Results on waterfall tracking are shown in Fig. 2. To make the illustration more clear, we show the tracking result of a single water point. In Fig. 2, the water point is tracked in eight successive frames before it disappears. In comparison to our work, other approaches (e.g., [15]) only calculate 2D velocity. They do not perform any point selection or impose further motion constraints. With these algorithms, accurate point tracking would be difficult.

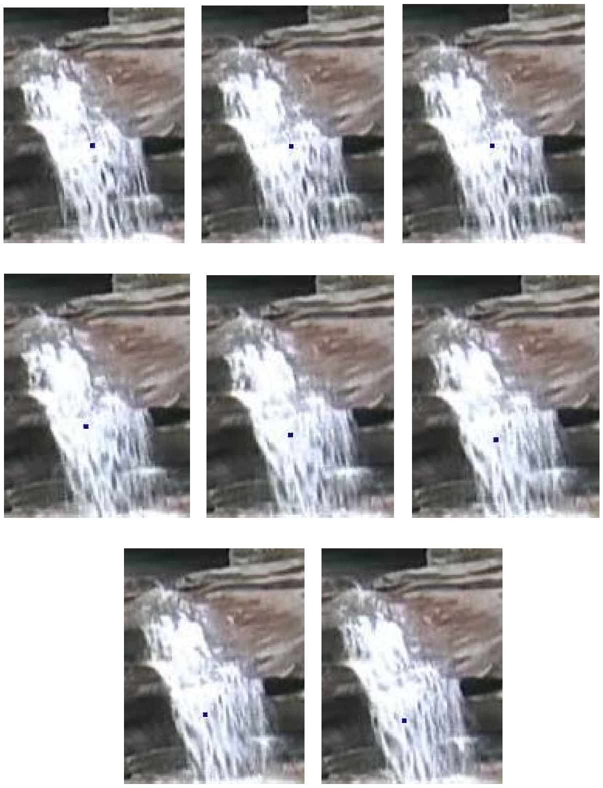

Fig. 2. Tracking of a single water point across eight video frames

In addition to motion separation and tracking, our optic flow based motion detection and analysis also provides means for motion modeling and synthesis. Using the rectified flow velocities, we extract different motion parameters for different waterfall regions in video sequences, which can be applied for scene animation.

IV APPLICATIONS IN ANIMATION

Optic flow and motion tracking can be used in many ways in animation. For example, the movement of snowflakes could be tracked, and a measure of the randomness in movements could be made to simulate a more natural snowfall. Similarly rainfall patterns could be modeled directly from analysis of a rainfall video. Complex scenes like waterfalls could be synthesized directly from video analysis, making it possible to model these events using only a few stills and motion vectors. Movements of airplanes could be tracked in a video, allowing a more natural representation of dynamic events.

Creating an animation, however, involves many steps, an example of which is outlined below. The animated short, “Dream Travel of a Paper Airplane”, was partitioned into six sub-scenes, each focusing on a predetermined subject. There are technicalities that are common amongst all the scenes, for example animating a moving background. We decided to use a paper airplane as our subject. The reason is that we could make our own paper airplane and take pictures of it at different angles as we needed, without having to search or model real airplanes in different viewpoints.

The next process for achieving image-based animation is to plan the set of scenes through which the plane should fly-through. Following are some of the scenes for our animation:

1) Folding and Take-off
   Stop Motion Animation is shown in this part. Some of the still frames captured in the process of building a paper airplane, and the final plane in different orientations are shown in Fig. 3. Intermediate frames were interpolated from these key frames.
2) Cliff
   Simulated erratic movement of the airplane at high velocity is shown in this part. To simulate this kind of motion starting with actual flow and motion tracking data, random fluctuations have been added to the flight trajectory that has been computed based on video flow analysis.
3) Desert and Weather
   Light, shadow and weather changes, such as storm, rainfall are shown in Fig. 4; normal desert scenes are shown on the top, while the scenes modified with storm, clouds and rain are shown on the bottom. The generation of rainfall and snowfall effects for background animation applies the same methodology of motion modelling and synthesis, as presented above for the case of waterfall motion.

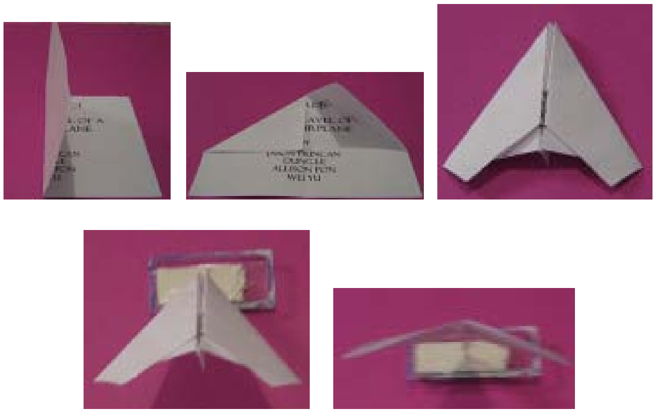

Fig. 3. Some of the still frames captured for animating plane.

The other scenes show the animation of Dusk Transition, Oasis as well as the final scene of the plane landing on a tree. A dramatic sunset is synthesized in the final scene transferring a sunset background on to a tree scene, as shown in Fig. 5. Shown in Fig. 6 are a few snapshots from the finally constructed animation sequence.

Screen shots from the animation of waterfall generated through flow detection and motion synthesis is shown in Fig. 7.

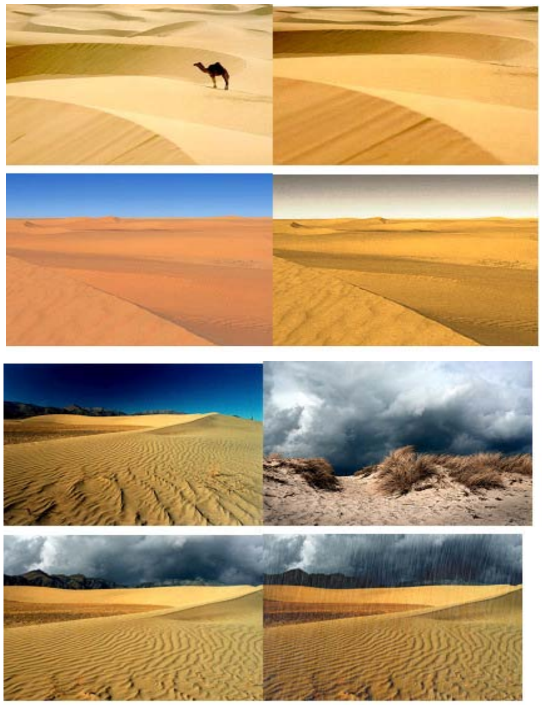

Fig. 4. Desert scenes: normal on the top and storm and weather effects on the bottom (Color Plate 3)

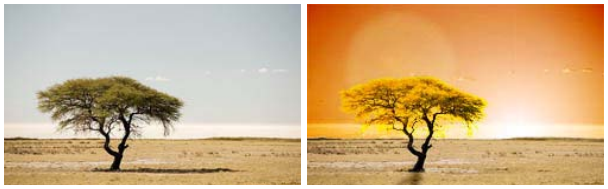

Fig. 5. Original tree scene on the left and sunset effect on the right

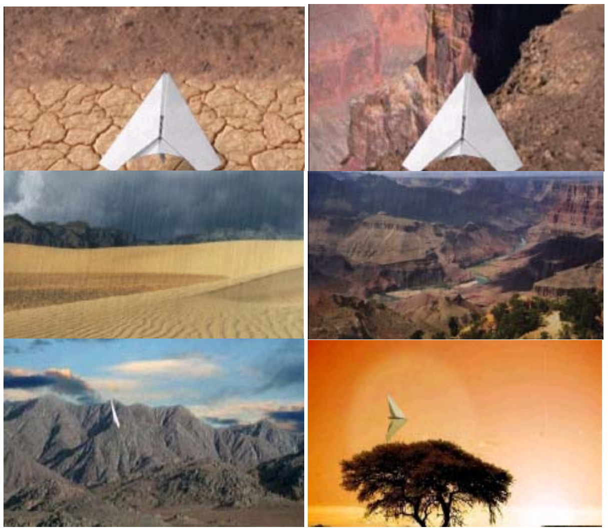

Fig. 6. Screenshots of different stages of paper plane animation (Color Plate 2)

Additional screenshots from another animation clip, showing season change and snowfall can be seen in Fig. 8. Flash demos and video of the animation can be viewed at www.cs.ualberta.ca/~lin/dream.

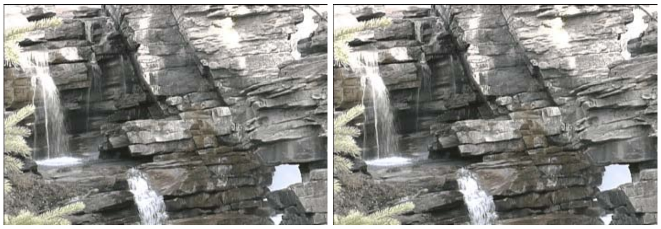

Fig. 7. Screenshots from waterfall animation

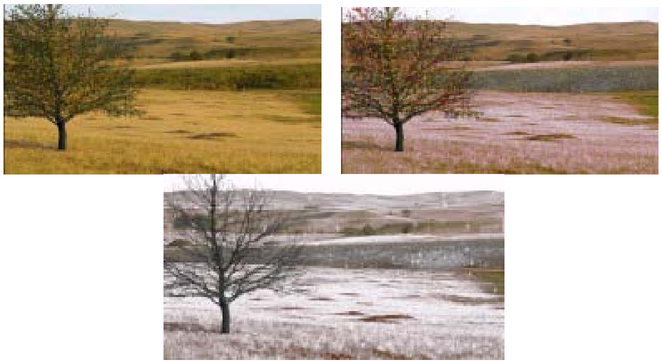

Fig. 8. Screenshots from season change and snowfall animation

V. CONCLUSION AND FUTURE WORK

In this work we outlined a method for robustly detecting optic flow and tracking feature point in videos of natural events, such as waterfall, rainfall and snowfall. A brief outline of the steps involved in creating simple animations with natural scenes was also given. In future work, we plan to model and track real airplanes to create more realistic video based animations. Also, we would like to integrate waterfall synthesis into a larger scene for animation.

ACKNOWLEDGEMENT

This work is funded by the Natural Science and Engineering Research Council (NSERC) of Canada; Informatics Circle of Research Excellence (iCore), Alberta; and the European Commission, for the research project µDrones.

The implementation work of the CROME project team is gratefully acknowledged. Special thanks are due to Dung Le, Jason Duncan, Wei Yu and Alison Pon for putting together the animation demonstration.

REFERENCE

[1] B.K.P. Horn and B.G. Schunck. Determine optical flow, A*rtificial Intelligence*, 17(1-3), pp. 185-203, Aug. 1981.

[2] T. Kanade and B. Lucas. An iterative image registration technique with an application to stereo vision, Proc. *IJCAI*, pp. 674-679, 1981.

[3] A. Bruhn, J. Weickert and C. Schnörr. Lucas/Kanade meets Horn/Schunck: Combining local and global optical flow methods, *Int. J. Computer Vision*, 61(3), pp. 211-231, Feb. 2005.

[4] C. Yuan, F. Recktenwald and H. A. Mallot. Visual steering of UAV in unknown environments, Proc. *IROS*, pp. 3906-3911, 2009.

[5] N.C. Gupta and L.N. Kanal. 3D motion estimation form motion field, *Artificial Intelligenc*e, 78(1-2), pp. 45-86, Oct. 1995.

[6] M.I.A. Lourakis, A.A. Argyros and S.C. Orphanoudakis. Independent 3D motion detection using residual parallax normal flow fields. Proc. *ICCV*, pp. 1012-1017, 1998.

[7] W.J. MacLean, A.D. Jepson and R.C. Frecker. Recovery of egomotion and segmentation of independent object motion using EM algorithm, Proc. *BMVC*, pp. 175-184, 1994.

[8] C. Yuan and H.A. Mallot. Visual motion analysis for 3D robot navigation in dynamic environments, *Robot Vision*, pp. 37-60, 2010.

[9] I.K. Park, H. Zhang and V. Vezhnevets. Image-based 3D face modelling system. *EURASIP Journal of Applied Signal Processing*, 2005(1), pp. 2072-2090, Jan. 2005.

[10] C. Theobald, M. Magnor, P. Schüler and H.P. Seidel. Combining 2D feature tracking and volume reconstruction for online video-based human motion capture, *IEEE Pacific Conf. on Computer Graphics and Applications*, pp. 96-103, 2002.

[11] L. Yin and A. Basu. Realistic animation using extended adaptive mesh for model based coding, Proc. *EMMCVPR*, pp. 269-284, 1999.

[12] J. Hays and I. Essa. Image and video based painterly animation, ACM Symposium on Non-Photorealistic Animation, pp. 113-119, 2004.

[13] S. Umeyama. Least-squares estimation of transformaiton parameters between two point patterns. *IEEE PAMI*, 13(4), pp. 376-380, April 1991.

[14] C. Yuan. Simultaneous tracking of multiple objects for augmented reality applications, Proc. *EGMM*, pp. 41-47, 2004.

[15] M. Black and P. Anandan. Robust dynamic motion estimation over time, *IEEE CVPR*, pp. 296-302, 1991.

Chunrong Yuan is a lecturer at the Cognitive Neuroscience Unit, University of Tübingen, Germany, where she leads a group carrying out research related to robot vision. She obtained in 1992 her M.Sc. in Computer Science & Engineering from Shanghai Jiaotong University and in 2004 her Ph.D in Computer Science from University of Erlangen-Nürnberg in Germany. Her research interests include Computer Vision, Computer Graphics, Human Computer Interaction for Virtual and Augmented Reality, and vision-based 3D navigation of autonomous Robots.

Irene Cheng is the Scientific Director of the iCORE-NSERC Multimedia Research Center and an Adjunct Faculty in the Department of Computing Science, University of Alberta, Canada. Her research interests, among others, include incorporating human perception following psychophysical methodology to improve multimedia transmission techniques. Recently she has designed a multimedia framework incorporating innovative interactive 3D item types for online multimedia education and led a team in implementing the Computer Reinforced Online Multimedia Education (CROME) System (http://crome.cs.ualberta.ca).

Anup Basu received his Ph.D in CS from University of Maryland, College Park, USA. He originated the use of foveation for image, video, stereo and graphics communication in the early 1990; an approach that is now widely used in industry standards. He also develops the first robust (correspondence free) 3D motion estimation algorithm using multiple cameras (IJCAI"87), a robust (and the first correspondence free) active camera calibration method (CVPR'93), a single camera panoramic stereo (ICPR'96), and several new approaches merging foveation and stereo with application to 3D TV visualization and better depth estimation. His current research applications include 3D/4D Image Processing and Visualization especially for medical applications, Multimedia in Education and Games, and Wireless 3D Multimedia transmission. He has been a Professor in the CS department of University of Alberta since 1999 and is currently an iCORE-NSERC Industry Research Chair.

Hanspeter A. Mallot is a professor and the founder of the Cognitive Neuroscience Unit, University of Tübingen, Germany. He received his Ph.D from the Faculty of Biology, University of Mainz, Germany in 1986. His research activities focus on spatial cognition in rats, humans and robots, using behavioral experiments in virtual reality, eye-movement recordings, software and hardware based simulated agents as well as autonomous robots. Currently he is the Dean of the University Faculty of Biology.