Experiments on Lecturer Segmentation using Texture Classification and a 3D Camera

Gerald Friedland, Kristian Jantz, Lars Knipping, Raul Rojas
Institut für Informatik
Freie Universität Berlin
[fland|jantz|knipping|rojas]@inf.fu-berlin.de

April 2005

Abstract

In our system for recording and transmitting lectures over the Internet the board content is sent as vector graphics, yielding a high quality image, while the video of the lecturer is sent as a separate stream. It is easy for the viewer to read the board, but the lecturer appears in a separate window. To eliminate this problem, we segment the lecturer from the video stream and paste his image on the board image at video stream rates. The lecturer can be dimmed by the remote viewer from opaque to semitransparent, or even transparent. This paper explains the two techniques we apply to achieve this: texture classification based segmentation, and segmentation using a novel 3D camera based on the time-of-flight of backscattered light principle. We argue that this technique provides a solution to the divided attention problem which arises when board and lecturer are transmitted in two different streams.

1 Introduction

Lectures held in front of a blackboard can be captured and transmitted in two ways: Either as a video of lecturer and board, or as a set of strokes and images captured by an electronic whiteboard which are rendered with high quality in the remote computer. In order to record or transmit classes, it has become common to use either standard Internet video broadcasting systems [36, 35, 23] or software that records and/or transmits stroke based information [28, 19]. The advantage of using state-of-the-art video broadcasting software is its availability and straightforward handling. The disadvantages are the high bandwidth and file storage capacity required [1]. Also, some video compression techniques used by the software can lead to deterioration of the board image[2].

[1] See for example [37]. A 90 minutes talk in MPEG-4 format, for example, can swell to 657 MB.

[2] DCT or Wavelet based codecs assume that higher frequency features of images are less relevant, and this produces an unreadable blurring of the board handwriting or a bad compression ratio.

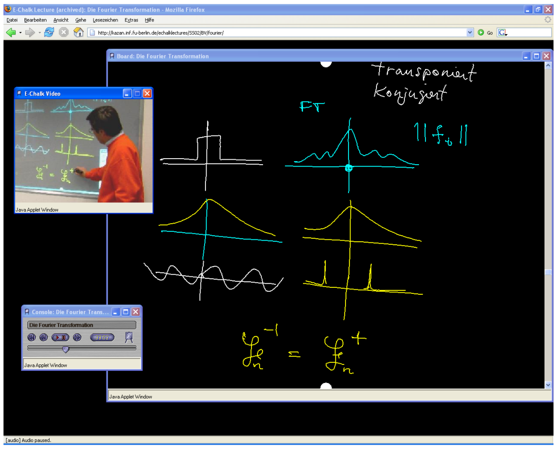
E-Chalk Lecture (archived): Die Fourier Transformation - Mozilla Firefox
Datei
Bearbeiten
Ansicht
Gehe
Lesezeichen
Extras
Hilfe
http://kazan.inf.fu-berlin.de/echalklectures/SS02/BV/fourier/
Go
Board: Die Fourier Transformation
transponiert
Konjugiert
FT
||f_t||
E-Chalk Video
Java Applet Window
Console: Die Fourier Trans...
Die Fourier Transformation
Java Applet Window
Java Applet Window
[audio] Audio paused.

Figure 1: Example of a remote lecture: the board image is transmitted independently of streaming video

Pen tracking devices, on the other hand, capture strokes that can be transmitted and rendered as a crisp image; the strokes can be further processed, for example, by handwriting recognition software [31]. However, when only the board image is transmitted, the mimic and gestures of the instructor are lost. For this reason, many lecture recording systems do not only transmit the slides or the board content but also an additional video of the instructor [11, 22] (compare Figure 1). However the issue of divided attention arises [2, 30]: We have two areas of the screen competing for the viewer's eye: the video window showing the instructor, and the board or slides window.

In our project, we cut the video image of the lecturer from the video stream separating it from the background. The image of the instructor can then be overlaid on the board, creating the impression that the lecturer is working directly on the screen of the remote student. Mimic and gestures of the instructor appear in direct relation to the board content. Moreover, the image of the lecturer can be made semi-transparent, to look through the lecturer, or opaque.

This article presents and compares two approaches we have tested to solve the lecturer segmentation problem: A texture based classification algorithm and a hardware solution using a time-of-flight 3D camera. The article first reviews some related work, we then discuss texture based segmentation, and finally, 3D camera based segmentation.

2 Related Work

The standard technology for overlaying foreground objects onto a given background is chroma keying [13]. This technique is not applicable to our segmentation problem, because the background of the scene is neither fixed nor monochromatic. Separating the foreground from either static or dynamic background is the object of current research, see for example [20]. Much work has been done on tracking objects for computer vision (like robotic soccer [33], surveillance tasks [17], or traffic applications [4]), and also on interactive segmentation for image processing applications [5]. Numerous computationally intensive segmentation algorithms are being developed in the MPEG community, for example [7]. In computer vision, real-time performance is more relevant than segmentation accuracy as long as the important features can be extracted from each video frame. For photo editing applications, accuracy is more important and algorithms can rely on information obtained through user interaction (see discussion in [32]). For the task we investigate here, the segmentation should be as accurate as possible and non-interactive. A real-time solution is needed for live transmission of lectures.

The use of stereo cameras for the reconstruction of depth information has been thoroughly investigated. Disparity estimation is a calculation intensive task. Since it involves texture matching, it is affected by the same problems as texture classification methods, that is, similar or homogeneous areas are very difficult to distinguish and real-time processing requires additional hardware [39].

3D laser scanners are being used with increasing frequency, for example for the conservation of historical heritage [27], special effects [10], and autonomous robots [24]. They use use triangulation to reconstruct depth information. This process of reconstructing the 3D model of an object is computationally expensive [3].

Göktürk and Tomasi [15] investigated the use of 3D time-of-flight sensors for head tracking. They use the output of the camera as input for various clustering techniques in order to obtain a robust head tracker.

As we can see from this short overview, researchers have concentrated on each of two possible alternatives: Segmentation of objects exploiting depth information and segmentation of objects using texture and/or motion cues. In our project we pursue both strategies.

3 Texture Based Segmentation

3.1 Setup

Our scenario is that of an instructor using an electronic board[3] in front of a classroom. Some software systems [12] record and transmit all actions on the board, while a video camera synchronously captures a video of the lecturer. Figure 2 shows a video of a lecturer.

The method described here was inspired by GrabCut [32]. While GrabCut requires a rectangular area to be specified by the user, our algorithm tries to learn a coarse separation of the foreground objects by exploiting the temporal differences between several

[3] Examples of such hardware can be found at [18, 25, 34].

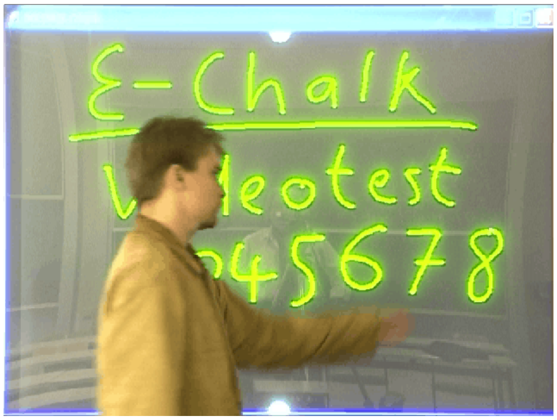

Figure 2: The image of the lecturer as captured with the video camera.

frames. Like GrabCut, the algorithm exploits color and color distribution information to improve the segmentation and better predict the spatial relationship of picture elements.

3.2 Temporal Foreground and Background Classification

The input for the classifier is a sequence of digitized YUV video frames. Each frame is subdivided into 8×8 pixel blocks. The classifier uses two main data structures:

- A foreground block buffer that is filled with any blocks that have a high chance of being part of the foreground, and
- a background buffer that contains those blocks classified as being definitely part of the background.

A block is moved into the foreground buffer if, during a sequence of n frames, the block has changed more than twice. Our experiments have shown that a good value is setting n to half the frame rate. A block is considered to have changed, when it differs significantly from the block at the same position in the previous frame, for example, according to the Euclidean distance. The background buffer contains all blocks that have never changed during the sequence being processed, and which were never classified as foreground during later operations. Both foreground buffer and background buffer are organized as ageing FIFO queues.

3.3 Color Distribution Classification

All frames are color quantized from YUV to a fixed 256 color palette (using 4 bits for Y and 2 bits for each U and V) and are divided into 8×8 pixel blocks. For each quantized block, the color histogram is calculated. The block histograms are now classified

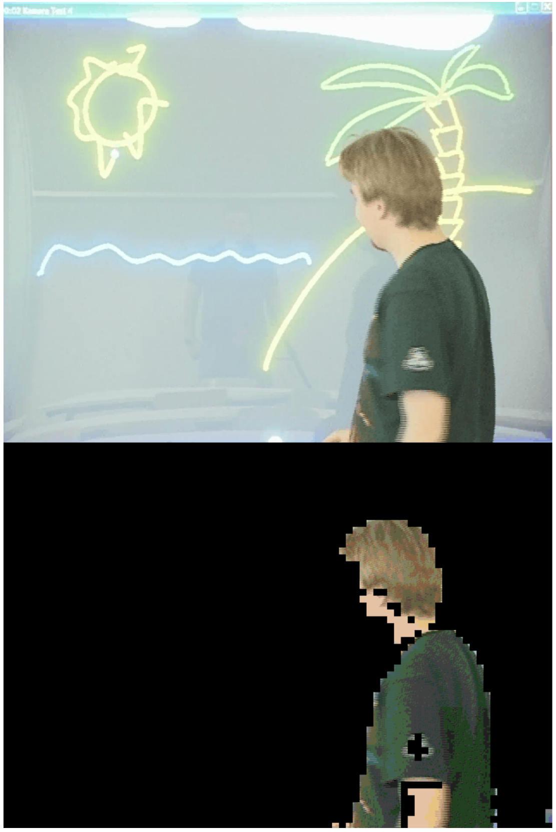

Figure 3: Original image (above) and output of the color distribution classification (below). The output improves, when a temporal classifier is used in addition.

into foreground and background by comparing each of them with block histograms of the foreground and background buffer. Metrics tested for comparison include the Euclidean distance and the Earth Mover's Distance [8]. A result of this classification applied to an image is shown in Figure 3.

3.4 Combining the Classifiers

The temporal classifier tends to find the borders of moving objects, while the color distribution classifier is better for surfaces. Given a frame and the results of the two classifications, any block considered foreground by at least one of the classifiers is considered foreground. The remaining blocks are a subset of the real background. For the foreground blocks, a connected component analysis is performed. The biggest blob is considered to be the instructor, and all other blocks (mostly noise or other moving objects) are put into the background buffer. Edge detection, using the Sobel Operator

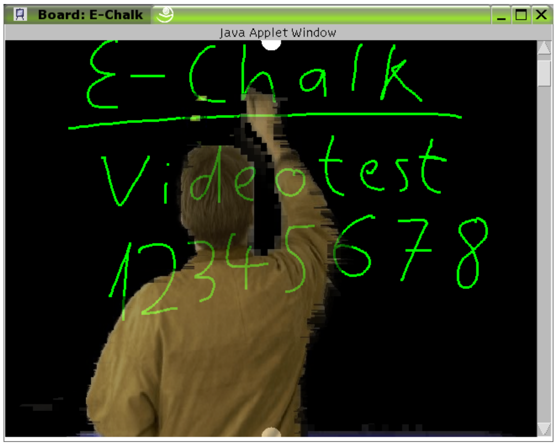

Figure 4: A lecturer extracted from the video stream and superimposed on the board image. The lecturer has been pasted as a semitransparent object.

[16], helps to smooth the edges of the blob, which appear ragged because of the resolution reduction to 8×8 pixel blocks. Smaller holes are filled and the corresponding block pixels are taken out of the background list. The resulting segmented video is scaled to fit the board resolution and is pasted over the board content at the receiving end of the transmission or lecture replay. Figure 4 shows the result. The current prototype implementation is still far from real-time performance[4] but we are sure this rate can be dramatically increased, for example, by processing only regions of interest, and by utilizing the SIMD multimedia instruction sets of modern CPUs.

4 Time-of-Flight Segmentation

Time-of-flight principle 3D cameras are now becoming available (see for example [9, 29, 6, 1]). They make real-time segmentation of objects easier, avoiding the practical issues resulting from 3D imaging techniques based on triangulation or interferometry. For our experiments we tested a miniature camera called SwissRanger SR-2 [9] built by the Swiss company CSEM, and a prototype camera called Observer 1K built by the German company PMD Technologies [29]. The cameras emit frequency modulated light in the infrared spectrum. This signal is backscattered by the scene and is detected by the cameras. An array of sensor elements is able to demodulate the signal and detect its phase, which is proportional to the distance to the reflecting object. The output of the cameras consists of depth images and conventional low-resolution gray scale video, as a byproduct. A detailed description of the time-of-flight principle can be found in [21, 26, 14]. Camera settings, such as frame rate, integration time, and user-defined region of interest (ROI) can be entirely configured by writing on internal registers of

[4] At this time we are able to process roughly one frame per second on a standard 2GHz PC.

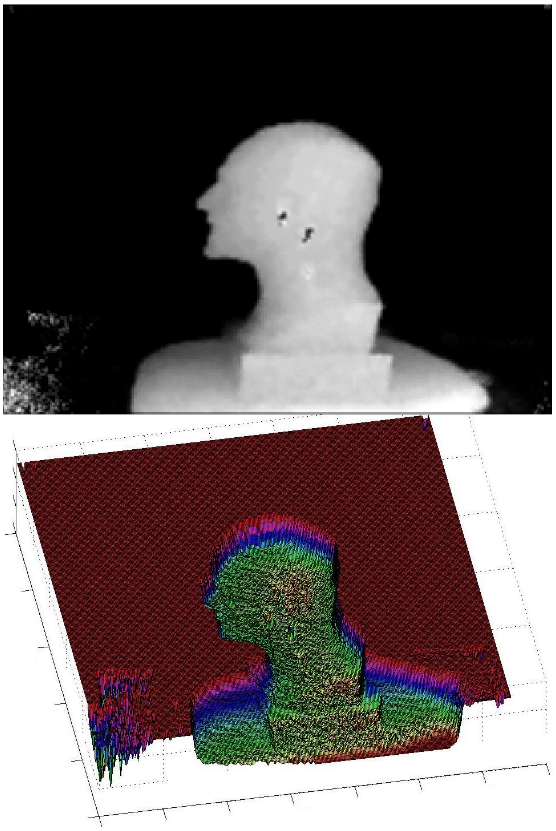

Figure 5: Depth-data represented by gray values (above) and 3D reconstruction (below). The resolution was quadrupled by fusing multiple images displaced half a pixel.

the camera. The resolution of the SwissRanger camera is 160×124 non-square pixels. The resolution of the Observer 1K is 64×16 non-square pixels. In a separate project the resolution could be doubled for each dimension at the cost of a lower frame rate by combining several pictures from slightly shifted camera positions (see Figure 5). Both cameras behaved similarly in our experiments, but its low resolution made the Observer 1K unusable for our purposes. We therefore concentrated on experimenting with the SwissRanger. The depth resolution depends on the modulation frequency. For our experiments we used a frequency of 20 MHz which gives a depth range between 0.5 m and 7.5 m, with an accuracy of about 1 cm.

4.1 Segmentation

The setup is similar to the scenario described in Section 3.1. The SwissRanger 3D camera was mounted on top of the video camera, and both cameras started capturing data synchronously. The 3D camera delivers images with depth information. We achieved a frame rate of about eleven frames per second. When an object is too close to the camera lense, overflows occur due to the large amount of light that is reflected. When an object is too far away from the lense, the distance measurement becomes imprecise. We found, that the optimal range for segmentation in terms of minimal visual noise and low overflow probability is between 2 m and 4 m. Ideally, the camera setup should be located at the end of the room to minimize the optical disparity between the video and the 3D camera without requiring a mirror setup. Due to the restricted range of 7,5 m, the disparity is noticeable. It is therefore necessary to calculate the disparity by knowing each camera's radial distortion and by scaling the captured 2D image according to the depth reported. Knowing the radial distortion and having fixed both cameras at the right position, segmentation reduces to setting a lower and upper bound for the depth values, and identifying corresponding pixels in the video stream. Only those pixels from the 2D video that are within the specified depth range are shown in the output. Figure 6 illustrates the result of segmenting a sphere from a video stream.

5 Comparison

Each approach we have investigated has its own set of advantages and disadvantages. The advantage of using texture based methods is that by choosing the right heuristics, it is possible to get useful results purely with software. Domain specific assumptions, like "the instructor area always begins at the bottom of the image", can improve the precision, as well as the speed of the segmentation. Texture based methods can be used without special hardware. Their disadvantage is the high computational cost incurred, and the method can fail when foreground and background textures are very similar. Interlacing effects, reflections, and shadows can produce problems, too. Segmenting areas with skin color (hands, faces) is especially difficult [38]. In our implementation, we track the instructor and do not handle yet the reinitialization needed when the lecturer moves out of the video frame. Due to motion blur, the edges are not as sharp as they should be. Still another problem is that if the instructor points at a rapidly changing object (for example, an animation on the board screen), the two corresponding blobs could become merged. However, such artifacts are not as distracting as it may seem, and we continue working on improving the quality of the lecturers image.

Using a time-of-flight 3D camera the computational costs are dramatically reduced. The 3D camera captures depth and intensity information at acceptable frame rates. The segmentation problem is theoretically reduced to a simple depth range check (compare Figure 7). However, the exact calibration and synchronization of the two cameras is tricky. The 3D cameras do not yet provide any explicit synchronization capability, such as those provided by many FireWire cameras. The low x and y-resolution of the 3D camera results in coarse edges. The z-resolution is just about enough, since the instructor stands usually very close to the board (and then the range of interest becomes about 50 cm). Besides overflows, there are other artifacts caused by quickly moving objects, light scattering, background illumination, or the non-linearity of the

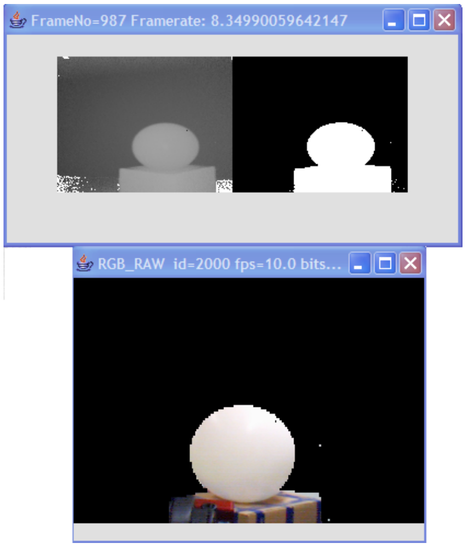
FrameNo=987 Framerate: 8.34990059642147
RGB_RAW id=2000 fps=10.0 bits...

Figure 6: Depth data (upper left), mask computed for a specific range (upper right), and mask applied to the video image (below). The sphere has been cut perfectly from the video stream. During calibration, the video and 3D camera have been matched correctly.

measurement. We also found that the depth measurement is not texture and material independent. Since darker objects reflect less light, the output of the camera is noisier than in the measurement of brighter objects. Last but not least, using a time-of-flight camera requires a larger budget, at least for now. For our purposes, the ideal time-of-flight camera should offer a higher depth range (for example 15 m) and a *z*-axis resolution of a few millimeters. The image resolution should be at least PAL. It would be ideal if a color video chip could be combined with the depth measurement chip in a single unit.

6 Conclusion

This paper proposes a novel solution to the divided attention problem which arises when board and lecturer are transmitted separately, in order to improve the quality of the board or slides reproduced at the receiving end. We propose to cut the lecturer out of the video stream and paste it on the rendered image of the board. Our experiments show that this approach is feasible and also esthetically appealing. The superimposed lecturer does not distract the student, it helps the student to better associate the lecturer's gestures with the board contents.

The paper presents and compares two approaches for instructor segmentation in board based lecture recordings. We presented a working example of a texture based seg-

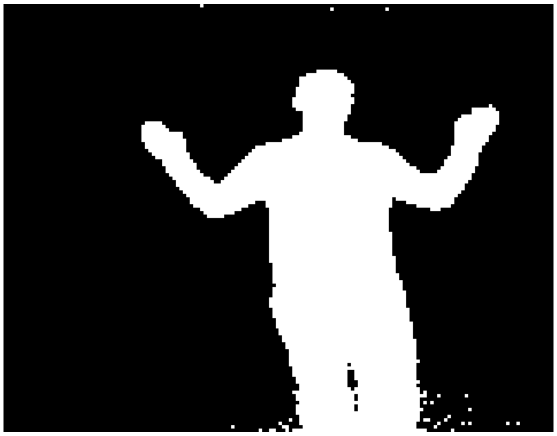

Figure 7: The mask obtained by depth range check for segmenting an instructor.

mentation algorithm and compared it to using a time-of-flight 3D camera. Although time-of-flight cameras are promising, they are not yet available with sufficient quality and resolution. The approach we want to follow in the immediate future is combining both methods: The 3D device could be used to get an approximation of the shape of the instructor, and a software based classifier can refine the shape to get a clean segmentation result. Therefore, 3D cameras will change the way certain segmentation tasks can be performed, but they will not eliminate 2D segmentation approaches in the near future. Their combination seems the best alternative to achieve our goal of mixing two data streams into a single high-quality one.

It is also worth noting, that although the methods presented in this paper cannot yet be applied in real time at 30 frames per second, the data can already be processed off-line, the result can be stored, and the lectures can be replayed using the superimposed lecturer technique described in this paper. Our future work will extend what can be already done off-line to real-time transmissions.

We thank Cüneyt Göktekin for the preparation of the high-resolution 3D images, and Gerald Weber for enlightening discussions about lecturer's segmentation.

References

[1] 3DV Systems Inc. DMC 100 Depth Machine Camera. http://www.3dvsystems.com, 2004.

[2] B. J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, Cambridge, UK, 1988.

[3] F. Bernadini, I. M. Martin, and H. Rushmeier. High-quality texture reconstruction from multiple scans. *IEEE Transactions on Visualization and Computer Graphics*, 7(4):318–332, October-December 2001.

[4] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A Real-time Computer Vision System for Measuring Traffic Parameters. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 1997.

[5] Y. Boykov and M.-P. Jolly. Interactive Graph Cuts for Optimal Boundary and Region Segmentation of Objects in N-D Images. In *Proceedings of the International Conference on Computer Vision*, pages 105–112, Vancouver, Canada, July 2001.

[6] Canesta Inc. CanestaVision EP Development Kit. http://www.canesta.com/devkit.htm, 2004.

[7] S.-Y. Chien, Y.-W. Huang, S.-Y. Ma, and L.-G. Chen. Automatic Video Segmentation for MPEG-4 using Predictive Watersheds. In *Proceedings of IEEE International Conference on Multimedia and Expo*, pages 239–243, Tokyo, Japan, August 2001.

[8] S. Cohen. *Finding Color and Shape Patterns in Images*. Ph.d. thesis, Stanford University, Department of Computer Science, 1999.

[9] CSEM Sa. SwissRanger 3D Vision Camera. http://www.swissranger.ch, 2004.

[10] Eyetronics Inc. Eyetronincs 3D Laser Scanner. http://www.eyetronics.com, 2005.

[11] G. Friedland, L. Knipping, and R. Rojas. E-Chalk Technical Description. Technical Report B-02-11, Fachbereich Mathematik und Informatik, Freie Universität Berlin, May 2002.

[12] G. Friedland, L. Knipping, J. Schulte, and E. Tapia. E-Chalk: A Lecture Recording System using the Chalkboard Metaphor. *International Journal of Interactive Technology and Smart Education*, 1(1), February 2004.

[13] S. Gibbs, C. Arapis, C. Breiteneder, V. Lalioti, S. Mostafawy, and J. Speier. Virtual Studios: An Overview. *IEEE Multimedia*, 5(1):18–35, January–March 1998.

[14] S. B. Göktürk, hakan Yalcin, and C. Bamji. A Time-Of-Flight Depth Sensor - System Description, Issues and Solutions. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Washington D.C., USA, July 2004.

[15] S. B. Göktürk and C. Tomasi. 3D Head Tracking Based on recognition and Interpolation Using a Time-Of-Flight Depth Sensor. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Washington D.C., USA, July 2004.

[16] R. Gonzalez and R. Woods. *Digital Image Processing*. Addison-Wesley, Boston (MA), USA, 1992.

[17] I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-Time Surveillance of People and Their Activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):809–831, August 2000.

[18] Interactive Whiteboards, Wireless Pads, and Digitizers. GTCo CalComp Peripherals. http://www.gtco.com/, 2004.

[19] C. Jesshope. Cost-effective Multimedia in Online Teaching. *Educational Technology and Society*, 4(3):87–94, 2001.

[20] L. Li, W. Huang, I. Y. H. Gu, and Q. Tian. Foreground Object Detection from Videos Containing Complex Background. In *Proceedings of ACM Multimedia 2003*, Berkeley, California, USA, November 2003.

[21] X. Luan, R. Schwarte, Z. Zhang, Z. Xu, H.-G. Heinol, B. Buxbaum, T. Ringbeck, and H. Hess. Three-dimensional intelligent sensing based on the PMD technology. *Sensors, Systems, and Next-Generation Satellites V. Proceedings of the SPIE.*, 4540:482–487, December 2001.

[22] M. Ma, V. Schillings, T. Chen, and C. Meinel. T-Cube: A Multimedia Authoring System for eLearning. In *Proceedings of E-Learn*, pages 2289–2296, Phoenix, Arizona, USA, November 2003.

[23] S. Mukhopadhyay and B. Smith. Passive capture and structuring of lectures. In *Proceedings of the seventh ACM international conference on Multimedia*, pages 477–487, Orlando, Florida, USA, October 1999.

[24] A. Nüchter, H. Surmann, K. Lingemann, and J. Hertzberg. Consistent 3D Model Construction with Autonomous Mobile Robots. In *Lecture Notes of Artificial Intelligence 2821*, pages 550–564, Heidelberg, Germany, 2003. Springer Verlag.

[25] Numonics Corperation. The Interactive Whiteboard People. http://www.numonics.com/, 2004.

[26] T. Oggier, M. Lehmann, R. Kaufmann, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc. An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger). *Optical Design and Engineering. Proceedings of the SPIE.*, 5249:534–545, Februar 2004.

[27] C. Ogleby. Laser Scanning and Visualisation of an Australian Icon: Ned Kelly's Armour. In *Proceedings of 7th International Conference on Virtual Systems and Multimedia*, pages 201–208, California, USA, October 2001. IEEE.

[28] E. Pedersen, K. McCall, T. Moran, and F. Halasz. Tivoli: an electronic whiteboard for informal workgroup meetings. In *Proceedings of the conference on Human factors in computing systems (INTERCHI)*, pages 391–398, Amsterdam, the Netherlands, April 1993. ACM Press.

[29] PMD Technologies GmbH. PMDTec 3D Vision Camera. http://www.pmdtec.com, 2004.

[30] J. Raskin. *The Humane Interface: New Directions for Designing Interactive Systems*. Addison-Wesley, Boston (MA), USA, 2000.

[31] R. Rojas, G. Friedland, L. Knipping, and E. Tapia. Teaching With an Intelligent Electronic Chalkboard. In *Proceedings of ACM Multimedia 2004, Workshop on Effective Telepresence*, pages 16–23, New York, New York, USA, October 2004.

[32] C. Rother, V. Kolmogorov, and A. Blake. GrabCut - Interactive Foreground Extraction using Iterated Graph Cuts. In *Proceedings of ACM Siggraph Conference*, August 2004.

[33] M. Simon, S. Behnke, and R. Rojas. Robust real time color tracking. In *RoboCup 2000: Robot Soccer World Cup IV*, pages 239–248, Heidelberg, Germany, 2001. Springer.

[34] Smart Technologies, Inc. Interactive Whiteboard Technology. http://www.smarttech.com/, 2004.

[35] Stanford University. Stanford Computer Science Lecture Recording. http://www.stanford.edu/class/ee380/, 2004.

[36] U. o. C. The Berkeley Multimedia Research Center. BMRC Lecture Browser. http://bmrc.berkeley.edu/projects/lb/, 2003.

[37] X. Wu. Videos of ACM Multimedia 2004 Panel Sessions. http://www.cs.columbia.edu/ xiaotaow/acmmm/, October 2004.

[38] Q. Zhu, C.-T. Wu, K.-T. Cheng, and Y.-L. Wu. An adaptive skin model and its application to objectionable image filtering. In *Proceedings of ACM Multimedia 2004*, pages 56–63, New York, New York, USA, October 2004.

[39] C. L. Zitnick and T. Kanade. A Cooperative Algorithm for Stereo Matching and Occlusion Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):675–684, July 2000.