Cut & Paste: Merging the Video with the Whiteboard Stream for Remote Lectures

Gerald Friedland
fland@inf.fu-berlin.de

Kristian Jantz
jantz@inf.fu-berlin.de

Raúl Rojas
rojas@inf.fu-berlin.de

May 2005

ABSTRACT

In the system we use for recording and transmitting lectures over the Internet, the board content is transmitted as vector graphics, producing thus 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 onto the board image at video stream rates. The lecturer can be dimmed from opaque to semitransparent, or even transparent. This paper explains the techniques we apply to achieve this and argue that it can also compete with state of the art image segmentation used for foreground extraction in still images. The approach does not only provide a solution to the divided attention problem which arises when board and lecturer images are transmitted in two different streams, it can also be applied to a variety of other problems where a foreground object must be segmented.

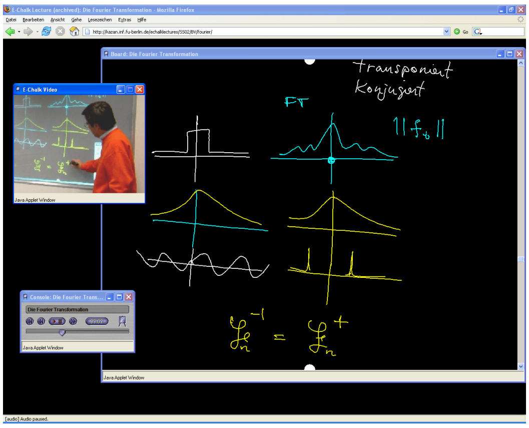
E-Chalk Lecture (archived): Die Fourier Transformation - Mozilla Firefox
Datei Bearbeiten Ansicht Gehe Lesezeichen Extras Hilfe
http://kazan.inf.fu-berlin.de/echalklectures/SS02/BV/fourier/
Go
Board: Die Fourier Transformation
transponiert
Konjugiert
FT
$\|f_t\|$
$\mathcal{F}_n^{-1} = \mathcal{F}_n^{+}$
E-Chalk Video
Console: Die Fourier Trans...
Die Fourier Transformation
33:02
Java Applet Window
[audio] Audio paused.

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

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 [38, 37, 26] or software that records and/or transmits stroke based information [29, 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] 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, using handwriting recognition software [32]. 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 [10, 23] (compare Figure 1). However the issue of divided attention arises [2, 31] because 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 in real-time from the steadily changing 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 now in direct correspondence to the board content. Moreover, the image of the lecturer can be made opaque or semi-transparent, in order to look through the lecturer.

This article presents the techniques we are using for real-time segmentation of the lecturer and argue that the proposed method, although relatively simple, can also compete with state-of-the-art image segmentation used for foreground extraction in still images, where no real-time constraints apply and usually more information is provided.

Although motivated by the divided attention problem which arises when board and lecturer are transmitted in two different streams, the approach presented here can also be applied to a variety of other problems where a foreground object has to be segmented.

The article first reviews some related work, we then present our segmentation approach for videos, and we show how it can be used for still images. We also briefly discuss two further methods we experimented with in order to make the segmentation even more robust.

2. RELATED WORK

The standard technologies for overlaying foreground objects onto a given background are chroma keying and background subtraction[13]. These techniques are not applicable to our segmentation

[1] See for example [39] 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.

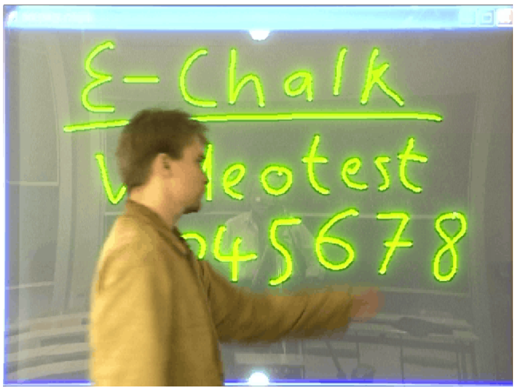

Figure 2: The image of the lecturer as captured with the video camera.

problem, because the background of the scene is neither monochromatic nor fixed. Much work has been done on tracking objects for computer vision (like robotic soccer [35], surveillance tasks [17], or traffic applications[4]). Most of these approaches concentrate on special features of the foreground and in these domains, real-time performance is more relevant than segmentation accuracy as long as the important features can be extracted from each video frame. Numerous computationally intensive segmentation algorithms have been developed in the MPEG community, for example [7].

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 [42].

Göktürk and Tomasi [14] investigated the use of 3D time-of-flight sensors for head tracking. They use the output of the 3D camera as input for various clustering techniques in order to obtain a robust head tracker.

Separating the foreground from either static or dynamic background is the object of current research, see for example [21]. Many systems use complex statistical methods that require intensive calculations not possible in real-time or use domain-specific assumptions [20]. Although non-parametric approaches exist [9], per-pixel Gaussian Mixture Models (GMM) are the standard tool for modeling a relatively static background [12]. In our scenario the background is constantly changing, while on the other hand the instructor sometimes stands still. This makes a clear distinction between foreground and background difficult. Therefore, we decided to concentrate on modeling the background using a representative sample of pixels.

For photo editing applications accuracy is more important than real-time performance and algorithms can rely on locality information obtained through user interaction [5]. For interactive still image segmentation, several algorithms exist (see for example the discussion in [33]). 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.

3. LECTURER EXTRACTION IN VIDEOS

Our principal scenario is that of an instructor using an electronic board[3] in front of a classroom. Our software system [11] records and transmits all actions on the board, while a video camera synchronously captures a video of the lecturer. It is assumed, that the camera views the scene directly from a distance where lense distortion is neglectable. Figure 2 shows a frame of such a recorded video. The segmentation approach consists of several steps. The overall idea is to use temporal differences in the video stream to determine what parts of the picture constitute background. The background is constantly changing while the instructor works on the chalkboard, but sometimes the instructor does not move at all. This makes it difficult to build a model of the background straight away. In most cases, however, there are several parts of the image which remain constant over a certain time. These parts are assumed to be a representative sample of the background. Using such representative colors for the background, a color classificator determines what parts of each frame belong to the foreground. The color classificator also contains a simple model of the board image in order to suppress the lecturer's writings and drawings. The biggest connected foreground component is assumed to be the instructor. Figure 3 shows an overview of the processing chain, which is explained next.

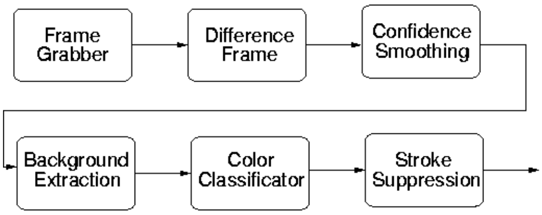

Figure 3: The main processing steps for lecturer extraction.

3.1 Exploiting Temporal Information

The input is a sequence of digitized YUV or RGB 640×480 pixel video frames either from a recorded video or directly from a camera. Each frame is converted to CIE-LAB [40] space, which approximates a perceptually uniform color space. The advantage is that the Euclidean distance between two colors in this space better approximates a perceptually uniform measure for color differences than in any other color space, like YUV, HSI, or RGB. To reduce the computational cost for this operation, we slightly smooth each band of the YUV or RGB color space and use a hash table to reuse already computed conversions. Using our experimental videos, the table grows to about 16 MB (because a recorded board video contains about one million different colors). Each processing step receives as input the CIE-LAB frame and a confidence matrix from the preceding classificator. The confidence matrix contains each pixel's probability of belonging to the foreground. The first processing step simply uses a Gaussian noise filter and calculates the difference of two consecutive frames pixelwise using Euclidean distance. The confidence matrix is initialized with these distance values normalized between 0 and 1.

The next processing step is to apply exponential smoothing on the last three confidence matrices. We found, that this improves the frame rate independence of the algorithm.

3.2 Reconstructing the Background

It is not trivial to build a model of the background since it is changing continously. The instructor not only writes on the board, the surface he or she is writing on sometimes reflects objects in the classroom. In a rear projection surface, the overall lightness of the surface changes with the writing color. In Figure 4 we illustrate a worst case example, which is very rare in practice. The instructor can paste images or even animations onto the board and when

[3] Examples of such hardware can be found at [18, 27, 36].

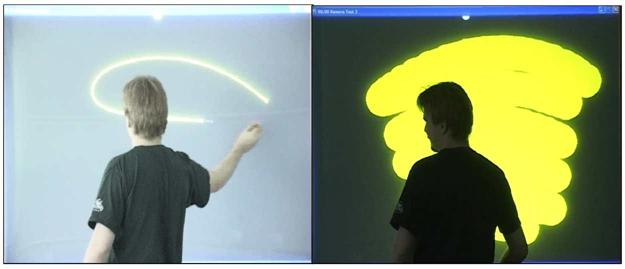

Figure 4: Worst case example of a change of lighting conditions during a lecture. The algorithm needs about a second to recover from this extreme example.

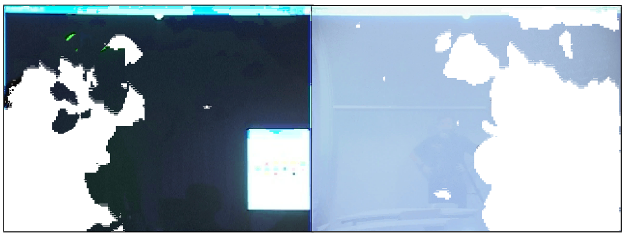

Figure 5: Two examples of reconstructed backgrounds. The white regions could not be identified because of too much movement. However, the identified regions give a representative statistical sample of color and texture of the background.

the instructor scrolls a page of board content upwards, the entire screen is updated. However, the instructor sometimes stands still producing less changes than the background noise. The idea is thus to extract only a representative subset of the background, that does not contain any foreground for further processing.

To distinguish noise from real movements, we use the following simple but general model. Given two measurements m_1 and m_2 of the same object with each measurement having a maximum deviation e of the real world due to noise or other factors, it is clear that the maximum possible deviation between m_1 and m_2 is $2e$. Given several consecutive frames, we estimate e to find out which pixels changed due to noise and which pixels changed due to real movement. To achieve this, we record the color changes of each pixel over a time period $h_{(x,y)}$ (where x and y specify pixel coordinates). We assume that during this interval, the minimal change should be one that is caused by noise. We then divide the frame into 16 subframes and accumulate changes in each subframe. Under the assumption, that at least one of these subframes was not touched by any foreground object, we then estimate $2e$ to be the average variation of the subframe with the minimal sum. We then join all pixels of the current frame with the background sample that during this history period $h_{(x,y)}$ did not change more than our estimated $2e$. The history period $h_{(x,y)}$ is initialized with one second and is continously increased for pixels that are seldom classified as background, to avoid that a still-standing instructor is added to the background buffer. Figure 5 shows some examples of reconstructed backgrounds. In our experiments, it took several seconds, until enough pixels could be collected to form a representative subset of the background. We call this time period the initialization phase. The background sample buffer is organized as an ageing FIFO queue.

3.3 Color Segmentation

The method described here was adapted from [34] who describes the use of color signatures and the Earth Mover's Distance for image retrieval. The idea behind our approach is to create a kind of color signature of the representative background sample and use it to classify the pixels in the image into those belonging to the signature and those not belonging to it. The representative background sample is clustered into equally sized clusters because in LAB space specifying a cluster size means specifying a certain perceptual accuracy. To do this efficiently, we use the modified two-stage k-d tree [3] algorithm described in [34], where the splitting rule is to simply divide the given interval into two equally sized subintervals (instead of using the median). In the first phase, approximate clusters are found by building up the tree and stopping when an interval at a node has become smaller than the allowed cluster diameter. At this point, clusters my be split in several nodes. Therefore, in the second stage of the algorithm, nodes that belong to several clusters are recombined. To do this, another k-d tree clustering is performed using just the cluster centroids from the first phase. We use a cluster size of $\gamma \cdot 0.66$ for the L axis and $\gamma \cdot 1.32$ for both the A and the B axis, where γ is a user defined accuracy factor.

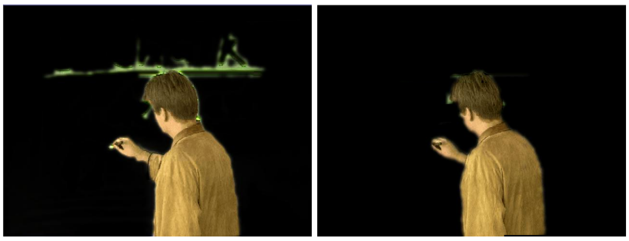

Figure 6: The result of board stroke suppression. Left image: without suppression; right image: with suppression.

For efficiency reasons, clusters that contain less than 0.2% of the pixels of the entire background sample are removed.

We explicitly build the k-d tree and store the interval boundaries in the nodes. Once built-up, the tree is only updated, when more than a quarter of the underlying background sample has changed.

Given a certain pixel, all that has to be done is to traverse the tree to find out whether it belongs to one of the background sample clusters or not. This allows for very efficient classification of the non-background pixels in each frame. The confidence matrix is then updated by averaging the results of the classification (1 for foreground, 0 for background) with the old confidence values. This lowers the risk, that colors that appear both in the background and foreground are classified in the end as background.

Figure 7 shows examples of the results of the color classification for moving images.

3.4 Suppressing Board Strokes

A connected component analysis is performed for the pixels classified as foreground - this means pixels with a confidence greater than 0.5. The biggest blob is considered to be the instructor, and all other blobs (mostly noise and other moving objects) are put back into the background buffer. In order to further suppress strokes drawn by the lecturer, all colors from the board system's color palette are inserted as cluster centroids to the k-d tree. However, as the real appearance of the writing varies with both projection screen and camera settings and with illumination, not all of the board activities can be suppressed. Additionally, strokes are surrounded by regions of noise that make them appear to be foreground.

Figure 6 compares two segmented frames with and without board writing suppression.

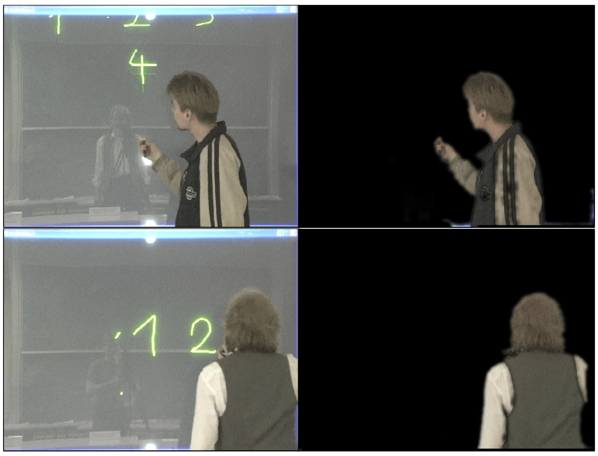

Figure 7: Two examples of color segmented instructors. Original frames are shown on the left, segmented frames are shown on the right. The above frame shows an instructor scrolling the board, which requires an update of the entire background. The frame below shows an instructor while he is writing.

3.5 Results

The elements of the confidence matrix are directly mapped to α-values, specifying the opaqueness of each pixel. Using a simple brightness filter helps to get rid of shadows, therefore they are not a major issue. Reflections on the board display are mostly classified as background and small moving objects are never classified as the biggest blob.

For the background reconstruction process to collect representative background pixels it is not necessary to record a few seconds without the instructor. The only requirement is, that for the first few seconds of initialization the lecturer keeps moving and does not occlude background objects that differ significantly from those in the other background regions.

The resulting segmented video is scaled to fit the board resolution (mostly 1024 × 768) and is pasted over the board content at the receiving end of the transmission or lecture replay. Figure 8 shows a result. Demonstration lecture replays can be found at *http://www.siox.org/videos/* .

The performance of the algorithm depends on the complexity of the background and on how often it has to be updated. Using our board segmentation videos, the current Java-based prototype implementation processes a 640 × 480 video at 4 frames per second. This includes a preview window and a motion JPEG compression. A 320 × 240 video can be processed at 12 frames per second on a standard 3 GHz PC. We are sure this rate can be dramatically increased, by utilizing the SIMD multimedia instruction sets of modern CPUs.

As the algorithm focuses on the background it provides rotation and scaling invariant tracking of the biggest moving object. The tracking still works when the instructor turns around or when he leaves the scene and a student comes up to work on the board. Once initialized, the instructor does not disappear, even if he stands still for several seconds.

4. SEGMENTATION OF STILL IMAGES

In order to test the robustness of the segmentation algorithm, we also developed an interactive version that works with still images. Instead of learning the background using temporal information, the user drags a rectangle with the mouse. The outside region of the rectangular area specifies the known background and the region inside the rectangle defines a superset of the foreground. Since it does not affect usability, the user can also specify one or more known foreground regions. This makes the classification more robust, as this lowers the probability that foreground colors that also exist in the background are classified as background - just like the mixing of the confidence values after the color classification in the motion based approach. Figure 9 shows the result of classifying the showcase picture in [33] and a the result of using Adobe Photoshop. The superset of the foreground is marked with the red rectangle and the green rectangles specify the optional known representative foreground regions.

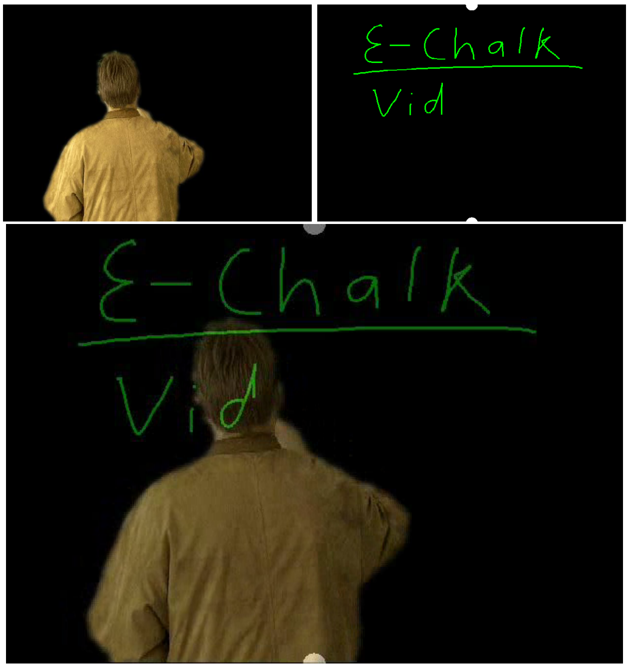

Figure 8: The segmented lecturer (above left) is superimposed semi-transparently on the vector based board data (above right) and replayed together either using a Java based client or using MPEG 4 (below).

Refer to their paper [33] for a detailed discussion and comparison of other state-of-the-art foreground segmentation approaches, like Graph Cut [5], or Intelligent Scissors [25]. To demonstrate the performance of our algorithm we have a demonstration Applet in the web at *http://www.siox.org/* , where one can choose between several of our typical segmentation video frames, the picture shown here, and images from the Berkeley Segmentation Benchmark Dataset [24].

Both, the LAB conversion table and the classification tree have to be build up from scratch for every still image segmentation. Therefore segmentation of still images needs more time than foreground extraction in consecutive video frames where the algorithm benefits from a reuse of the data structure. Using a standard 3 GHz PC our Applet segmentates pictures with a resolution of roughly 450 × 350 pixels in 0.5 to 1.5 seconds, depending on the complexity (amount of colors) of the image and the information provided by the user.

5. FURTHER EXPERIMENTS

Although this approach works surprisingly well, we have already

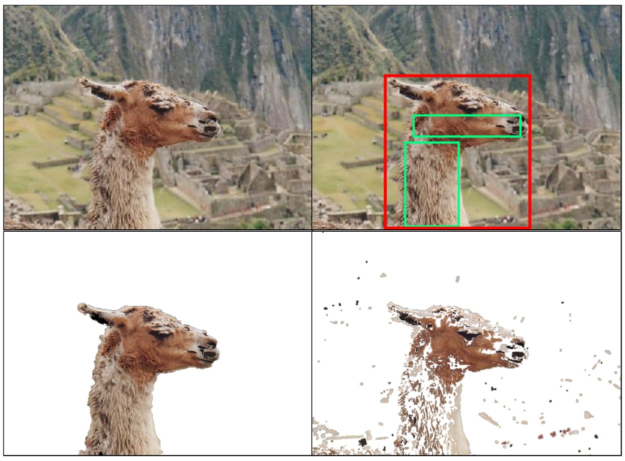

Figure 9: Testing our color segmentation approach with still images: The original image (above left), the user provided selection (above right), our result (below left), the result of using Adobe Photoshop's Magic Wand. For a detailed comparison of segmentation approaches on this picture, please refer to [33].

experimented with two additional methods, which could supplement the main technique to further improve the robustness of the segmentation.

5.1 Texture 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 into foreground and background by comparing each of them with block histograms in the background buffer using an approximation of the Earth Mover's Distance.[4] A result of this classification applied to an image is shown in Figure 10. The method is still far away from working in real-time. Using histograms requires the division of each frame into sub frames, which results in blocky artifacts that require the usage of additional filters to eliminate them. Used as an additional classificator, however, it improves the robustness of the foreground extraction because not only color information but also color distribution is taken into account. This helps to differentiate, for example, between green grass and a T-shirt that consists of a uniform green that lies in the same cluster.

6. TIME-OF-FLIGHT CAMERA SEGMENTATION

Time-of-flight principle 3D cameras are now becoming available (see for example [8, 30, 6, 1]). For our experiments we tested a miniature camera called SwissRanger SR-2 [8] built by the Swiss company CSEM. The camera emits amplitude 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 [22, 28, 15]. The resolution of the SwissRanger camera is 160 × 124 non-square pixels. The depth

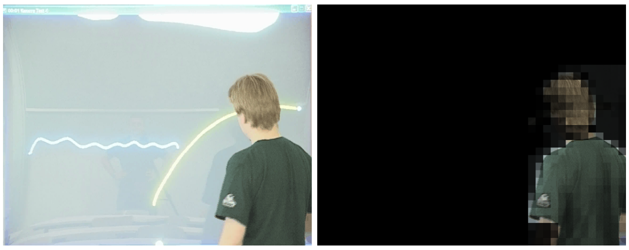

Figure 10: Original (left) and the result of an experimental texture classifaction on an instructor (right). Aside from the missing skin (due to color quantization), the transparency of the blocks is proportional to the amount of pixels that belong to the instructor.

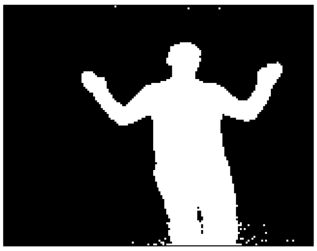

Figure 11: The mask obtained by depth range check for segmenting an instructor, as computed by us using a 3D camera.

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.

The 3D camera captures depth and intensity information at acceptable frame rates. As [16] already showed, range information can be used to get a better sample of the background faster. Moreover, the segmentation problem is theoretically reduced to a simple depth range check (compare Figure 11). However, the exact calibration and synchronization of the two cameras is difficult. 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 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.

7. CONCLUSION AND FUTURE WORK

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.

[4] We thank Yvonne Schindler, who is currently doing a master thesis on fast implementations of the Earth Mover's Distance.

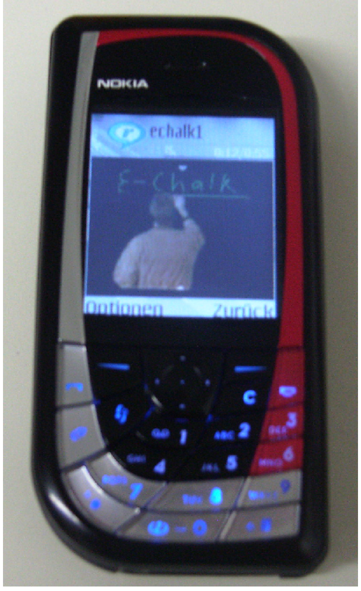

Figure 12: A 90 minutes lecture using dynamic board replay, audio and superimposed lecture can be stored and replayed on modern mobile devices, since it needs roughly 40 MB of storage.

Our experiments show that this approach is feasible and also esthetically appealing. The superimposed lecturer helps the student to better associate the lecturer's gestures with the board contents. Pasting the instructor on the board also reduces space and resolution requirements. This makes it possible to replay a lecture on a mobile device that contains also a video of the lecturer (see Figure 12). A lecture containing board, overlaid instructor and audio can be replayed on a mobile device at 64 kbit/s.

This paper presents a runtime efficient adaptation of image retrieval techniques to solve foreground segmentation problems in videos and still images. The method enables scale and rotation invariant tracking of foreground and also handles the classification of newly introduced objects. The presented approach has been kept general with only a very few domain specific assumptions. It can be applied to a variety of other problems where a foreground object should be extracted but only a partial reconstruction of the background is possible.

Like all color and/or texture based methods the approach can fail when foreground and background colors are too similar. Segmenting areas with skin color (hands, faces) is especially difficult [41]. 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 to improve the domain specific modeling of the board background. We also presented two further experiments we conducted, that improve robustness on the cost of additional resources (computational power and/or special hardware).

8. ACKNOWLEDGMENTS

We thank Christian Zick for helping us to record many videos for segmentation and for the in-depth testing of Photoshop's Magic Wand algorithm, and our former colleague and co-creator of E-Chalk Lars Knipping for the quick adaptation of the Java board client to make superimposing of the instructor possible for live transmissions.

9. 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] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18:509–517, 1975.

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

[9] A. Elgammal, D. Harwood, and L. Davis. Non-parametric Model for Background Substraction. In *Proceedings of the 7th IEEE International Conference on Computer Vision, IEEE ICCV99 Frame Rate Workshop*, Kerkyra, Greece, September 1999.

[10] 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.

[11] 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.

[12] N. Friedmann and S. Russel. Image Segmentation in Video Sequences: A Probablistic Approach. In *Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence (UAI97)*, Providence, Rhode Island, USA, August 1997.

[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 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.

[15] S. B. Göktürk, H. 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.

[16] G. Gordon, T. Darrel, M. Harville, and J. Woodfill. Background estimation and removal based on range and color. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, USA, June 1999.

[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] S. Jiang, Q. Ye, wen gao, and T. Huang. A New Method to Segment Playfield and its Applications in Match Analysis in Sports Video. In *Proceedings of ACM Multimedia 2004*, pages 292–295, New York, New York, USA, October 2004.

[21] 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.

[22] 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.

[23] 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.

[24] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[25] E. Mortensen and W. Barrett. Tobogan-based Intelligent Scissors with a Four Parameter Edge Model. In *Proceedings of the IEEE Conference on Computer Vision and pattern Recognition*, volume 2, pages 452–458, 1999.

[26] 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.

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

[28] 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.

[29] 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.

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

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

[32] 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.

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

[34] Y. Rubner, C. Tomasi, and L. J. Guibas. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[35] 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.

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

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

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

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

[40] G. Wyszecki and W. S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. John Wiley and Sons, New York, NY, 1982.

[41] 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.

[42] 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.