REAL-TIME SEGMENTATION FOR ADVANCED DISPARITY ESTIMATION IN IMMERSIVE VIDEOCONFERENCE APPLICATIONS

Ingo Feldmann, Serap Askar, Nicole Brandenburg, Peter Kauff, Oliver Schreer

Heinrich-Hertz-Institut, Berlin
Einsteinufer 37
10587 Berlin
Germany
{feldmann, askar, brandbg, kauff, schreer}@hhi.de http://bs.hhi.de/

ABSTRACT

We present an advanced disparity estimation algorithm developed for real-time immersive teleconferencing. In order to improve the disparity maps in the case of occlusions a segmentation-driven disparity post processing is applied. We introduce a novel approach for a segmentation algorithm, which uses the advantages of HSV colour space without the need of an explicit colour space transformation. This makes segmentation very fast while still keeping the robustness of HSV colour space especially to illumination changes and shadows. Further on, we present an algorithm for tracking the hands of a conferee. Finally, we introduce an algorithm framework, which combines segmentation, tracking of hands and disparity estimation to create robust disparity map in real-time. We show the improvement of results for image-based rendering.

Keywords: disparity analysis, segmentation, colour space, recursive block matching, real-time

1. INTRODUCTION

Immersive tele-presence systems, which are known for some time from the experimental laboratories, become more and more applicable at reasonable costs for daily use in tele-communication. The crown jewels of this evolution are immersive video-conferencing systems. Such systems provide a shared virtual environment where conferees at different geographical places can meet under similar conditions as in real world [Kauff00],[Schre01a]. To collaborate in the most effective and natural manner (see Fig.1) the participants are enabled to make use of rich communication modalities as similar as possible to those used in face-to-face meetings (e.g., gestures, eye contact, etc). This eliminates the limits of conventional tele-conferencing applications, (e.g., face-only images in separate windows, unrealistic avatars).

To realize such immersive video-conferencing systems 3D images of the conferees are captured and positioned consistently around a shared

Vision of immersive videoconference
Figure 1

position (determined by a head tracker). This scene is rendered onto a 2D display by an image-based rendering system. For this purpose, the virtual camera of such a system must correspond to the head of the conferee [Kauff00]. Moreover, the generation of realistic 3D video objects requires a multi-view camera set-up capturing the conferees. The disparities, which represent the depth of the video object, need to be estimated between corresponding images. Based on these disparity vector fields virtual views can be synthesized.

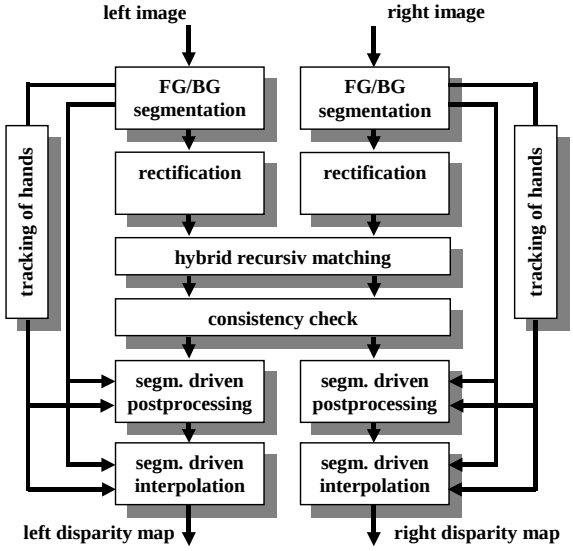
A lot of disparity estimation algorithms have been proposed for this purpose in context with stereo applications [Fauge93], [Alvar00], [Pritc98]. Several real-time stereo systems have been built around a parallel-camera configuration [Berto98], [Ohm98]. However, they do not meet the given real-time requirements concerning wide-baseline stereo with strongly convergent cameras. Therefore, we have recently proposed a new algorithm which is based on so-called hybrid recursive matching (HRM) - an universal approach on real-time analysis of displacement vector fields [Kauff01b], [Kauff01a], [Schre01b]. Here, HRM is able to provide sparse disparity fields at 8x8 or 4x4 grids related to full ITU-Rec. 601 resolution in real-time at state-of-the-art Pentium processors.

However, disparity analysis in 3D video conferencing requires more than a straightforward application of HRM only. The crucial problem is to produce an accurate and correct perspective view of the presented conferee in the virtual scene. Especially, occlusions caused by the hands, arms or head may result in visible artefacts in the final synthesis of the virtual view. The same holds for self-occlusions at object borders yielding image regions, which are only seen by one of the two stereo cameras. To cope with these special occlusion problems, the sparse disparity maps of the HRM algorithm have to be post-filtered by dedicated segmentation-driven algorithms. Those segmentation algorithms have to be robust to global illumination variations and shadows.

The improvement of results of an existing HRM disparity estimation algorithm is the main topic of this paper. In section 2 we introduce a sophisticated algorithm framework, which combines the disparity estimation with an advanced segmentation algorithm and a system for the tracking of the conferees hands. Both algorithms will be discussed in detail in what follows. Section 3 describes our novel approach for a robust real-time segmentation. It outlines the overall algorithm as well as our investigations of an adaptive threshold buffer system and an efficient post processing and refinement strategy. Section 4 discusses in detail the problem of illumination changes and shadows. We present a novel approach for a fast HSV colour space approximation in YUV, which we use for robust shadow detection and segmentation of skin colour. Section 5 presents the algorithm for tracking and segmenting the conferee's hands. Then, section 6 and section 7 describe how these segmentation data are used during post-processing of sparse disparity maps and interpolation of dense disparity fields. Finally, section 8 discusses some experimental results and section 9 summarises our work.

2. REAL-TIME DISPARITY ESTIMATION FRAMEWORK

In this section we describe our real-time disparity estimation framework. Fig.2 gives an overview. The overall goal is to estimate and correct dense-field disparity maps from the left to right and right to left camera input image[1]. The shown algorithm framework is separated into several modules. Each of them provides real-time dense-field results in ITU-Rec. 601 resolution. All modules run on a fast Dual Pentium IV system supported by multiple PCI based multi processor boards containing four TriMedia TM 1300 on each board.



Real-time disparity estimation
Figure 2

The first module is the foreground-background segmentation for both input images. It aims to separate the conferee from the scene background. This module is essential for the final view renderer[2], which integrates conferees from different locations seamlessly into a common virtual scene. Moreover, the segmented video objects in this way contain only about half of the pixels of the entire video frame. As a consequence, the computational complexity of subsequent processing steps is much smaller.

The hand tracking modules generate additional segmentation masks of conferees hands. They are based on skin colour segmentation, which is applied to bounding boxes surrounding and tracking hands in each image. The resulting masks of

[1] We need both maps for the consistency check and view matching in case of occlusions.
[2] Not shown in the figure

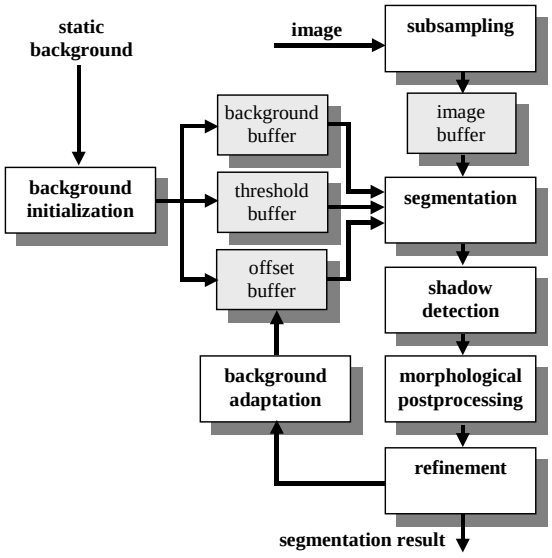
segmentation and hand tracking modules are used to improve the results of the hybrid disparity estimation algorithm in the post-processing step. This is essential especially in case of occlusions within the object. In the case of videoconference applications this especially applies to the hands of the conferees.

The hybrid recursive matching algorithm is introduced in detail in [Kauff01a]. It provides robust, spatially and temporally homogeneous disparity maps at a sparse grid. Due to its strict recursive structure, it may produce some mismatches, especially in occlusion areas. This is the tribute we have to pay for the real-time constraint. It is the task of the post-processing algorithm to detect these mismatches by an efficient consistency check and to substitute inconsistent disparities by adequate inter/extrapolation processes. This last step can also be utilised to derive dense disparity maps from the sparse fields.

Our algorithmic framework is a trade-off between the quality of result and the real-time constraint. Both, the disparity estimation and the segmentation work with subsampled data. In case of segmentation we downsample the image by a factor of 4. We use a subsequent refinement algorithm for upsampling within the segmentation module. It performs an additional segmentation of the objects borders in full resolution while binary interpolating the centre parts of the detected foreground objects straight away. The subsequent rectification works on full resolution (for more details see [Kauff01a]). The resulting images contain the rectified scene foreground (i.e. the conferee) only. The disparity estimation works on a 4×4 sparse grid again. The segmentation-driven interpolation of disparity data uses a similar interpolation and refinement strategy as the segmentation. Although we need to spend some effort in the refinement and interpolation our results show that the presented algorithmic framework is much faster by maintaining a very good quality of result compared to processing without subsampling at all.

3. SEGMENTATION

The structure of our segmentation module is shown in Fig.3. The segmentation algorithm is based on a change detection scheme, comparing the current image of a particular conferee with a pre-known scene background reference (see Fig.5) by a simple euclidic distance operation. The background reference buffer is captured during an initialisation phase at the beginning of the conference session where a short background sequence is analysed. This provides further statistical information about noise. From this we derive a pixel wise threshold buffer, which is used for segmentation. During the

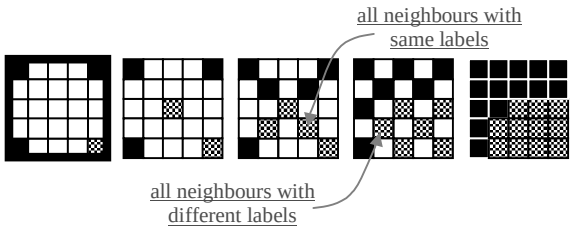


Block diagram of the proposed segmentation algorithm
Figure 3:

conference session we compensate global illumination changes and improper threshold values with an adaptive offset buffer.

The result of segmentation is a binary mask marking each pixel as foreground (i.e. object) or background. Because this mask is usually corrupted by noise we apply different kinds of well-known non-linear and morphological post-processing algorithms[3] to smooth the binary mask and get one closed object region, which represents the conferee (Fig.7). A major problem for segmentation are shadows. Therefore, we introduce a special shadow detection module (see section 4), which compensates such distortions. Fig.6 demonstrates the result of foreground-background separation with and without this module.

As mentioned in section 2 we subsample our input image by a factor of 4 to speed up the application. Therefore we use a combination of



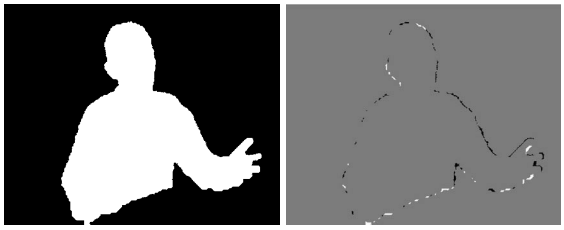
Hierarchical refinement strategy for a "transition block"
Figure 4

[3] Binary median filtering and subsequent open-close filtering

Background (left) and image with conferee (right)
Figure 5

YUV foreground/background segmentation without (left) and with (right) shadow detection
Figure 6:

YUV segmentation mask without (left) and with (right) morphological
Figure 7



HSV segmentation mask (left) and difference to YUV segmentation mask (right)
Figure 8

interpolation and refinement to upsample the final segmentation mask back to full resolution. This process is performed block wise. In homogenous areas, i.e. completely inside or outside the object region, the binary mask can be interpolated by simply copying the known pixels. At the transition regions segmentation must be particularly repeated in full resolution to preserve the contour as much as possible. This is achieved with a hierarchical filling strategy for the so called “transition blocks” as shown in Fig.4. During several recursive processing steps all missing pixels labels are filled. Therefore all neighbours of the according pixel are analysed first in a 4×4 grid, then in a 2×2 grid and finally in full resolution. If the majority of the neighbours have the same value the according pixel gets this value too. Otherwise for this pixel a new segmentation is performed. We have tested this algorithm to be very efficient and fast. One of its major features is that it always results in a closed region and that it does not produce any holes or sparkles in the interpolation transition area. Thus, the need of further post-processing is avoided.

Finally, based on the result of refinement the threshold offset buffer is updated as mentioned before.

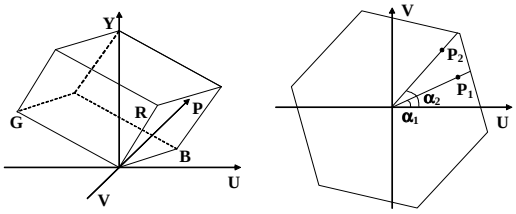
4. SHADOW DETECTION

Global illumination changes are usually smooth and occur over the whole image. They are eliminated by the adaptive threshold buffer system explained in section 3. In this section we concentrate on shadow detection. In difference to global illumination changes shadows may produce local faulty areas as shown in Fig.6. They occur due to the intensity shifts caused by the shadows. If those shifts exceed a given threshold for the difference based change detection they are wrongly interpreted as foreground. To overcome this problem we use a basic property of shadows. This is that in case of non-reflecting diffuse surfaces a shadow causes a change of intensity and saturation of a pixel's colour value only [Skarb94]. The hue will remain constant within reasonable bounds for saturation and intensity. Therefore, a segmentation based on hue only is robust to shadow distortions.

On the other hand, saturation and intensity provide useful additional information for segmentation. So, for example, foreground objects having the same hue as background but different saturation and intensity cannot be detected in a purely hue-oriented approach. They would be interpreted as shadows, i.e. background. So, an efficient segmentation algorithm should combine all available information. For example, an useful information is that shadows always make the image darker, i.e. reduce intensity and saturation. Exploiting this, we may scan only the "darker" pixel for equal hue. Further on, an additional threshold may limit the intensity shift caused by the shadow. Our results show that a careful adjustment of thresholds minimizes the failure and usually produce very good results (see Fig.6).

Our approach is a two-step method. As explained in section 3, first a conventional segmentation based on thresholding a simple

Euclidic distance is used. In this first separation between foreground and background a shadow will be interpreted as foreground due to its high intensity difference (Fig.6, left). In a second step only the obtained foreground region is scanned for shadows. This significantly reduces the amount of operations and makes the application much faster.



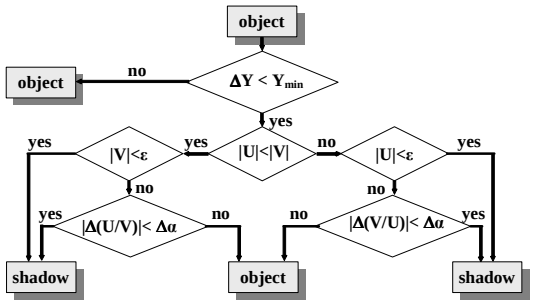
Representation of RGB values in the YUV colour space (left); Hue is represented by its angle to the u-axis, $\Delta\alpha$ is used as the segmentation threshold (right).

Figure 9

Because TV and video data are usually available in YUV colour space sophisticated colour space transformations are necessary to extract information about hue and saturation. Those transformations aim to provide a more uniform and accurate representation of colour interpreting it in the same way as the human perceptual system does. Such colour spaces have been studied extensively in the literature. In our work we have investigated the HSV colour space, which represents colour in terms of hue, saturation and intensity[4]. A detailed discussion is behind the scope of this paper. The interested reader may be referred to [Smith97]. In several tests we proved the HSV colour space to be robust to distortions caused by shadows. Fig.8 (left) shows an example for the resulting segmentation mask.

A problem of using different colour formats for real-time applications is the time-consuming colour space transformation. Therefore we refer to conventional chromakey techniques, which operate in the YUV colour space straight away[5] [Lang78]. Those techniques define a fixed background colour as reference and analyse the difference to this reference colour only. The YUV colour space does not provide absolute values for hue or saturation. But the change of a colour may be approximated. Considering a ray between a point P_i and the origin (see Fig.9, right) hue may be interpreted as the angle of that ray to the u-axes and saturation as the distance to origin [Lang78]. Two points P_1, P_2 are interpreted to have the same hue if their angles are equal and the same saturation if their distance to origin is equivalent. In case of segmentation a shadow (as a change of luminance, i.e. intensity) would shift a point P_i down on its ray to origin but not change the angle (see Fig.9, right). Therefore, the angle difference α_1-α_2 and a threshold value $\Delta\alpha$ may be used for the segmentation. In difference to chromakeying in our segmentation algorithm we do not use a fixed colour as reference but the one of the corresponding pixel in the background buffer.

A major drawback is still the computational effort for the trigonometric arctan function to obtain the angles α_i. The arctan(x) is defined by a first order Taylor series as follows:



Optimized Shadow Detection Scheme for YUV

Figure 10

$$\arctan(x) = \sum_{k=0}^{1} (-1)^k \frac{x^{2k+1}}{2k+1} = x - \frac{1}{3}x^3 \qquad (1)$$

For $|x| < 1$ it may be approximated by

$$\arctan(x) \approx x\,. \qquad (2)$$

For $|x| > 1$ we get: $1/|x| < 1$ and may approximate:

$$\arctan\left(\frac{1}{x}\right) \approx \frac{1}{x}, \quad \left|\frac{1}{x}\right| < 1 \qquad (3)$$

In Fig.10 we introduce our decision scheme for the whole shadow detection algorithm. As explained above we check only foreground pixel for equal hue (i.e. shadow). First, the luminance difference $\Delta Y = Y_{backgorund} - Y_{image}$ is limited by a upper threshold Y_{min} to avoid difference values above zero[6]. Because a shadow always reduces the intensity Y_{image} the difference ΔY must be less than zero for object assignment. Afterwards we choose

[4] Different to YUV colour space, which defines colour in terms of luminance Y and chrominance UV

[5] Video data are usually represented in YUV space

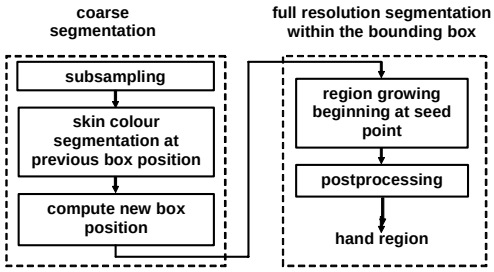
[6] In this case an analysis of hue and saturation makes no sense.

the appropriate arctan approximation by comparing U and V. Either, V or U are limited by a second threshold ε to limit the intensity shift caused by a shadow as explained previously. Finally, we choose $\Delta\alpha$ as threshold for hue.

The simplicity of our analysis algorithm makes it very fast. We tested the segmentation quality to be very similar to the much more complicated HSV colour space transformation (see Fig.8).

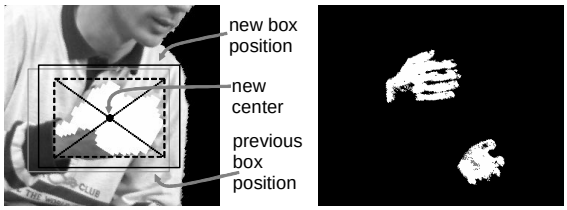
5. TRACKING OF HANDS

The key problem of hand tracking is to find a closed binary mask of the conferee's hands. The algorithm is separated into two subsequent steps. First, a bounding box is segmented and tracked over time for each hand in order to limit the search region and to speed up the whole processing. In the second step, a region growing technique is applied starting at a seed point (the centre of the hand) to extract closed regions of the hands. The diagram in Figure 11 shows the complete procedure.



Block diagram of the hand segmentation
Figure 11

The tracking of the bounding box is performed on a subsampled image. Using the segmentation algorithm explained in section 4 a hand blob is determined inside the bounding box. In difference to foreground-background segmentation in this case the current pixel's hue value is compared with a reference value, which represents the human skin colour within a wide range. The centre of the segmented hand blob delivers the position of the new bounding boxes and also provides the seed point for full resolution refinement (see Fig.12). This is performed by a region-growing algorithm, which follows a processing queue starting at the seed point [Mehne97]. For this purpose the seed point is admitted as first entry into the queue. Then, all four neighbours of queued pixel positions are analysed and those neighbours that fit in certain limits to the skin colour are added to the queue as well, whereas the others are excluded. Thus, the hand region grows in an iterative manner from the seed point up to the

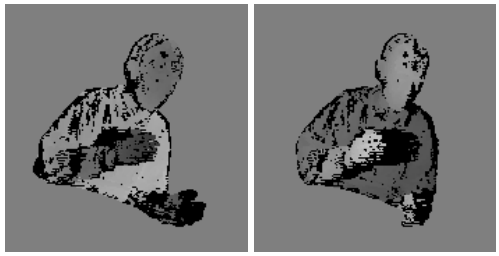


Hand blob and bounding boxes (left) and resulting binary hand masks (right)
Figure 12

boundaries. The results of this process are closed binary masks in full pixel resolution representing the left and right hand of the conferee (see Fig.12).

6. SEGMENTATION-DRIVEN POST-PROCESSING OF SPARSE DISPARITY MAPS

As already mentioned in section 2, the HRM-based processing step from Fig.2 produces a 4x4 sparse disparity map grid in pixel resolution. Due to the simple real-time structure of the algorithm, mismatches may occur in regions of homogenous texture as well as in occluded areas at the borders of the conferee's shape or in regions next to the arms and hands in front of the body. To cope with these particular matching problems (as well as to match views properly), the HRM algorithm is carried out twice, once for L→R and once for R→L disparity fields. This redundancy is utilised for a subsequent consistency check detecting the mismatches and removing inconsistent disparities at sparse grid level, while retaining the reliable disparity vectors.



Sparse disparity fields (L→R and R→L) after consistency check
Figure 13

In principle the consistency check simply compares the corresponding vectors of both disparity fields, taking into account that in ideal case a pair of corresponding disparities should have a zero-valued difference Δ. Thus, if the difference Δ is greater than a predefined threshold the according disparities are removed from the sparse field. A result of the consistency check is shown in Fig.13. The rejected disparities are marked in black.

The rejected disparities can roughly be classified in two categories. In the first category mismatches are caused by ambiguities in weakly textured or homogeneous regions. Disparities of the second category are rejected due to occlusions where image correspondences cannot be found as a matter of principle. Thus, filling of remaining holes in the sparse field requires quite different strategies. Those holes caused by poor texture can be filled by a simple interpolation between next reliable horizontal and vertical disparities. In contrast, missing disparities in occluded regions cannot be interpolated as they imply a discontinuity between two depth layers, which should be retained. Hence, in this case, missing disparities must be extrapolated from their neighbours in the same depth layer. This inter/extrapolation procedure is guided by the results of object/background separation from section 3 as well as hand tracking from section 5. In a first step, all direct neighbours who are inside the object mask are selected. Usually, this refers to the four neighbours from one of the disparity maps (see Fig.13). However, in holes at the object border the number of valid neighbours may be less than four, because some of them are outside the object. Here, the segmentation mask of object/background separation is used to determine the current number of inside neighbours.

Then, in a second step, it is tested whether or not the disparities of the selected inside neighbours are almost in the same range. If they are, it is assumed that the detected inconsistency has been caused by unstructured textures and not by occlusions. Thus, the missing disparity is simply substituted by an average from the selected neighbours, i.e. it is interpolated. If the selected neighbour disparities are not in the same range, the algorithm switches over to extrapolation incorporating the result from hand tracking and segmentation. If at least one of the selected neighbours belongs to a hand mask, the missing disparity is extrapolated from those neighbours, which belong to the same region; i.e. hand or non-hand region. Finally, if none of the selected neighbours is inside the hand masks, the missing disparity is taken from the next neighbour with the most similar texture.

Fig.14/15 show a comparison between simple interpolation and segmentation-driven extrapolation. The area around the left hand demonstrates the positive effect of using segmentation in combination with extrapolation. The smearing of disparities caused by interpolating in the hand-occluded area is avoided and the depth discontinuity at the right border of the hand is retained.

7. SEGMENTATION-DRIVEN INTERPOLATION OF DENSE DISAPRITY FIELDS

The goal of this module is the interpolation of a dense disparity field. We want to avoid the same problems as in case of post processing of disparity maps in section 6. These are artefacts at the object boundaries and at transitions between two different depth layers. Therefore, we take advantage of a segmentation-driven inter/extrapolation strategy again. The algorithm follows the hierarchical refinement approach of section 3. The only difference is that we use grey level images instead of binary masks. Therefore, new labels have to be bilinear interpolated from their neighbours instead of just being copied. If four neighbouring disparities have nearly the same value (see Fig.4), then it is assumed that the area within the window is of homogeneous depth. But whenever at least one of the four corner disparities significantly differs from the others, an extrapolation guided by hand segmentation masks and original texture data is applied. If the hand mask crosses the current block, the desired values are extrapolated from those disparities, which belong to the same depth layer. Otherwise, if no hand mask is available, the missing disparities are taken from that neighbour with the most similar texture. The final result is a dense disparity field in pixel resolution.

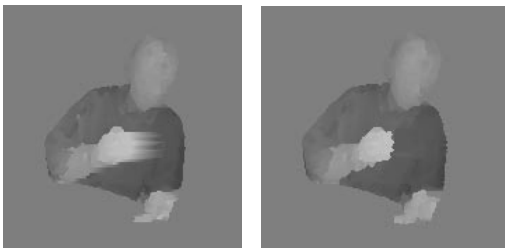
8. EXPERIMENTAL RESULTS

In our real-time environment, a fast Dual Pentium IV system is supported by multiple PCI based multi processor boards containing four TriMedia TM 1300 on each board. With this hardware extension it is possible to run all algorithms in real-time accepting a delay of less than 160 ms, i.e. four frames.

Fig.14 shows estimated disparity maps with and without a segmentation-driven disparity correction. Fig.15 presents the result of a virtual view synthesis as a close-up of the critical region around the hand. Both figures demonstrate that most of the artefacts could be eliminated. The depth discontinuity has been preserved and synthesized with good quality. Further on, our segmentation masks and disparity maps are robust to distortions caused noise or shadows and global illumination changes (see section 4).

9. CONCLUSION

In this paper we have discussed the improvement of the results of our HRM disparity estimation algorithm by a segmentation-driven post processing and interpolation of disparity maps. Our main focus was the improvement of segmentation as a trade off to the real-time constraint. We have presented an



Comparison of simple interpolation (left) and segmentation-driven extrapolation (right) for post-filtering of sparse disparity maps
Figure 14

Virtual view of conferees hand using a simple HRM (left) and segmentation based strategy (right)
Figure 15

effective segmentation module, which is robust to distortions, caused by noise or illumination changes and shadows. Further on, we introduced a fast hand tracking system, which helps to detect regions of self-occlusion during a conference session. Our algorithms work in real-time and produce robust results in full TV resolution.

ACKNOWLEDGEMENTS

This study is supported by the Ministry of Science and Technology of the Federal Republic of Germany, Grant-No.01 AK 022. Furthermore the authors would like to thank the European IST Project VIRTUE for the provision of the test sequences and for fruitful collaboration on the immersive tele-conference scenario.

REFERENCES

[Alvar00] L. Alvarez, R. Deriche, J. Sanchez and J. Weickert: „Dense Disparity Map Estimation Respecting Image Discontinuities: A PDE and Scale-Space Based Approach“, *INRIA Research Report No. 3874*, INRIA, Sophia-Antipolis, January 2000.

[Berto98] M. Bertozzi, A. Broggi: „GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection“, *Trans. on Image Processing*, Vol.7, No.1, January 1998.

[Fauge93] O.D. Faugeras et al: „Real-time Correlation-Based Stereo: Algorithm, Implementations and Applications“, *INRIA Research Report No. 2013*, INRIA, Sophia-Antipolis, August 1993

[Kauff00] P. Kauff, R. Schäfer, and O. Schreer: "Tele-Immersion in Shared Presence Conference Systems", *Proc. of IBC 2000, International Broadcast Convention*, Amsterdam, September 2000

[Kauff01a] P. Kauff, N. Brandenburg, M. Karl, O. Schreer: "Fast Hybrid Block- and Pixel-Recursive Disparity Analysis for Real-Time Applications in Immersive Tele-Conference Scenarios", *Proc. of WSCG 2001, 9th Int. Conf. on Computer Graphics, Visualization and Computer Vision*, Pilzen, Czech Republic, February 2001

[Kauff01b] P. Kauff, O. Schreer, J.-R. Ohm: "An Universal Algorithm for Real-Time Estimation of Dense Displacement Vector Fields", *Proc. of Int. Conf. on Media Futures*, Florence, Italy, May 2001

[Mehne97] A. Mehnert, P. Jackway: “An improved seeded region growing algorithm”, *Pattern Recognition Letters*, Vol.18, 1997, pp.1065-1071.

[Lang78] H. Lang: "Farbmetrik und Farbfernsehen", *R. Oldenburg VerlagVerlag GmbH*, Munich, 1978

[Ohm98] J.-R. Ohm et al: „A Real-Time Hardware System for Stereoscopic Videoconferencing with Viewpoint Adaptation“, *Image Communication, Special Issue on 3D-TV*, January 1998

[Pritc98] P. Pritchett and A. Zissermann: "Wide Baseline Stereo Matching", *Proc. of Int. Conf on Computer Vision*, Bombay, India 1998

[Schre01a] O. Schreer, P. Kauff: "An Immersive 3D Videoconferencing System Based on a Shared Virtual Table Environment", *Proc. of Int. Conf. on Media Fututres*, Florence, Italy, May 2001

[Schre01b] O. Schreer, N. Brandenburg, P. Kauff: "Real-Time Disparity Analysis for Applications in Immersive Tele-Conference Scenarios - A Comparative Study", *Proc. of ICIAP 2001, 11th int. Conf. on Image Analysis and Processing*, Palermo, Italy, Sept. 2001

[Skarb94] W. Skarbek, A. Koschau: "Colour Image Segmentation - A Survey -", *Technical Report 94-32, FB 13*, Technische Universität Berlin, October 1994

[Smith97] J. R. Smith: "Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis.", *Ph.D. thesis, Graduate School of Arts and Sciences*, Columbia University, New York, February 1997.