FAST HYBRID BLOCK- AND PIXEL-RECURSIVE DISPARITY ANALYSIS FOR REAL-TIME APPLICATIONS IN IMMERSIVE TELE-CONFERENCE SCENARIOS

Peter Kauff, Nicole Brandenburg, Michael Karl, Oliver Schreer

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

ABSTRACT

This paper presents a fast disparity analysis approach based on a hybrid block- and pixel-recursive matching scheme. The key idea is to choose efficiently a small number of candidate vectors in order to reduce the computational effort by simultaneously achieving spatial and temporal consistency in the resulting disparity map. The latter aspect is very important for 3D videoconferencing applications, where novel views of the conferee have to be synthesised in order to provide motion parallax. For this application a processing of video in ITU-Rec. 601 resolution is required. Our algorithm is able to provide dense disparity vector fields for both directions (left-to-right and right-to-left) in real-time at one Pentium III, 800 MHz processor in reasonable quality.

Keywords: disparity analysis, recursive block-matching, pixel-recursive matching, real-time, epipolar geometry, rectification.

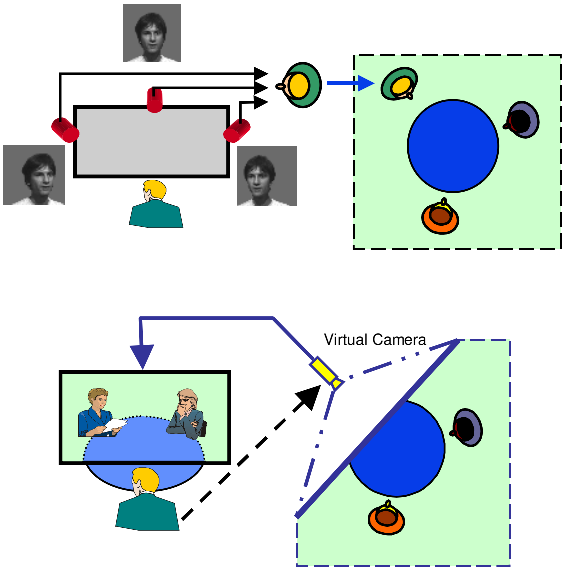
1. INTRODUCTION

Due to the fast progress in display and processor technology as well as in leading-edge signal processing related to computer vision and video coding, 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 tele-conferencing systems where conferees can meet in a shared virtual environment under similar conditions as in the real world [Schäfer00]. Here, immersive tele-presence means that the conferees will have the impression of being immersed in a virtual meeting room, sitting around a shared virtual table next to each other, and collaborating in the most effective and natural manner (see Figure 1).

To achieve this goal, 3D images of the conferees are captured and positioned consistently around the shared virtual table as shown in Fig. 2 (top).

Vision of immersive tele-conference
Figure 1

This virtual 3D scene is then rendered onto the 2D display of the terminal by using a virtual camera. The position of the virtual camera coincides with the current position of the conferee's head. For this purpose the head position is permanently registered by a head tracker and the virtual camera is moved with the head. Thus, supposing that all geometrical relations of capturing, compositing and rendering are well fitted, it is ensured that all conferees always see the scene under the same realistic conditions, especially while changing knowingly the view in order to watch the scene from another perspective, to look behind objects or to look at a previously occluded object - an aspect which is called parallax viewing and which is one main key issue of immersive tele-presence.



3D capturing (top) and rendering (bottom) for immersive tele-conference systems
Figure 2

To generate realistic 3D video objects, the conferees are captured by a multi-view camera set-up and disparities, which represent the depth of the video objects, are estimated between corresponding images. The virtual views can then be synthesised on the basis of disparity vector fields. A lot of disparity estimation algorithms have been proposed for this purpose in context with stereo applications. Historically, we can distinguish between two different methods, hierarchical block matching [Faugeras93] and optical flow algorithms [Barron94]. While block matching usually generates sparse disparity vector fields on block basis, using the minimisation of a certain cost function as a matching criterion, the optical-flow principle exploits the continuity between spatial gradients and differences of intensities between corresponding pixels in the two images. It usually produces dense vector fields.

However, most of these approaches do not meet the requirements of the immersive tele-conference scenario. This is mainly because of the following three reasons. First of all, the algorithms must be able to process full resolution video according to ITU-Rec. 601 in real-time - most desirable as pure software solution running on available processors without any support from dedicated hardware. Secondly, the disparity estimator should provide dense vector fields of high accuracy in order to guarantee a virtual view synthesis of adequate quality. And - last but not least - immersive tele-conference systems obviously have to use strongly convergent camera configurations due to the large size of immersive displays and the short distance between conferee and display. Therefore, real-time algorithms, known from literature and optimised for the simplified geometry of parallel or weakly convergent cameras [Bertozzi98], [Ohm98], can not be utilised for this application scenario.

To overcome these shortcomings, a new fast disparity estimator (FDE) which meets the above requirements is presented in this paper. It is based on a hybrid recursive matching algorithm which has already been applied successfully to fast motion estimation in format conversion and MPEG coding [Ohm97],[Smolic98],[Kauff00]. The main idea of this baseline algorithm is to unite the advantages of block-recursive matching and pixel-recursive optical flow estimation in one common scheme, leading to a fast hybrid recursive estimation algorithm, which will be explained in more detail in the next sections. To utilise this baseline algorithm for disparity estimation and to exploit the well-known epipolar constraint of stereo application in this context, the algorithm has been modified in such a way that it is able to search correspondences along epipolar lines. Alternatively, it can be used in combination with a preceding rectification which generates parallel views by applying a 2D transform to the convergent views [Fusiello97],[Robert97],[Schreer00].

In the next paragraph, we briefly review the epipolar geometry and the process of rectification. A short description of the new FDE algorithm follows. Then, we explain the details of the FDE algorithm such as block- and pixel-recursion as well as post-processing. Finally, experimental results will be presented and discussed.

2. REVIEW ON EPIPOLAR GEOMETRY

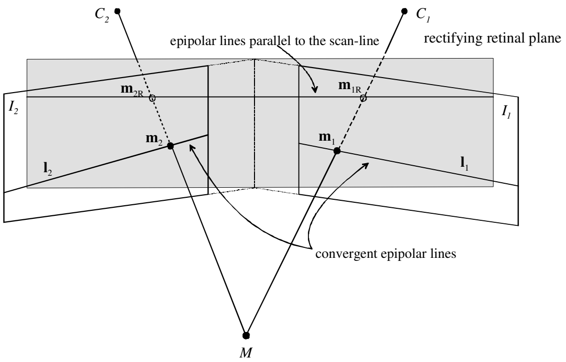
Starting from a 3D point M and its projections $\mathbf{m_1}$ and $\mathbf{m_2}$ onto the two image planes I_1 and I_2 of a stereo rig, the epipolar geometry tells us that the optical ray passing through $\mathbf{m_1}$ and M is mapped onto a corresponding epipolar line $\mathbf{l_2}$ in I_2 and that therefore $\mathbf{m_2}$ must lie on $\mathbf{l_2}$ if it is visible and not occluded in the second view (see Figure 3). Vice versa - $\mathbf{m_1}$ necessarily lies on the complementary epipolar line $\mathbf{l_1}$. This basic relation is described by the well-known epipolar equation where $\mathbf{F}$ denotes the fundamental matrix [Zhang96].

$$\tilde{\mathbf{m}}_1^{\,T}\mathbf{F}\tilde{\mathbf{m}}_2 = 0 \qquad (1)$$

Hence, the matching of corresponding points $\mathbf{m_1}$ and $\mathbf{m_2}$ can always be reduced to a 1-dimensional search along epipolar lines which are calculated as follows for each of the two available views:

$$\mathbf{l}_1 = \mathbf{F}\tilde{\mathbf{m}}_2 \quad \text{and} \quad \mathbf{l}_2 = \mathbf{F}^T\tilde{\mathbf{m}}_1 \tag{2}$$

Supposing a general stereo set-up with strongly convergent cameras, the 1-dimensional search can be implemented in a twofold manner [Schreer00]. The first procedure is a 1-step solution where the 1-dimensional search is directly carried out along arbitrarily oriented epipolar lines. The second one is a 2-step solution where both cameras are at first virtually rotated until they would represent a system with parallel stereo geometry. This pre-processing step is called rectification and generates pre-warped images with horizontal epipolar lines. Hence, point correspondences in the rectifying image planes can be searched along horizontal scan lines (see $\mathbf{m_{1R}}$ and $\mathbf{m_{2R}}$ in Figure 3). This clearly simplifies the implementation of the 1-dimensional search algorithm, but it also costs some extra computational load for the rectification itself.

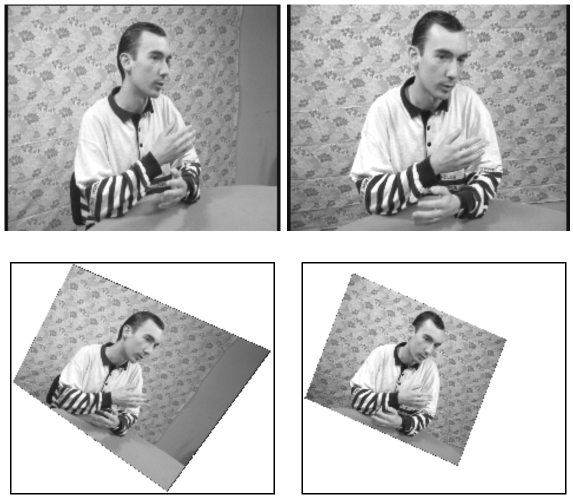


Epipolar geometry and rectified image planes
Figure 3

The image warping process representing the virtual camera rotation of rectification requires the derivation of two transformation matrices $\mathbf{T}_1$ and $\mathbf{T}_2$ from the camera geometry. To obtain these transformation matrices, a number of supplementary conditions are defined, leading to an unique solution of a homogenous system of equations [Fusiello97], [Robert97]. The resulting matrices can then be used to transform each pixel of the original view into a point in the rectified view.

$$\tilde{\mathbf{m}}_{1\mathrm{r}} = \mathbf{T}_1 \cdot \tilde{\mathbf{m}}_1 \quad \text{and} \quad \tilde{\mathbf{m}}_{2\mathrm{r}} = \mathbf{T}_2 \cdot \tilde{\mathbf{m}}_2 \tag{3}$$

Figure 4 shows an example for this rectification process. The two top images depict the two original views of a strongly convergent camera set-up which fits into the tele-conference scenario sketched in Figure 2 (cameras in right and middle position). The corresponding images of the rectified views are shown at the bottom.

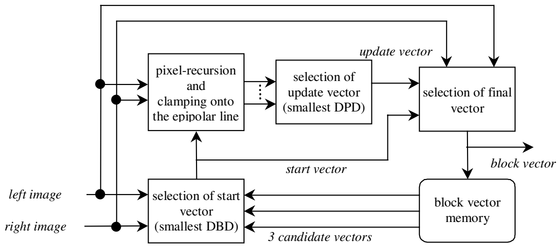


Original (top) and rectified (bottom, down-scaled) views of a strongly convergent camera set-up
Figure 4

3. OUTLINE OF THE FDE ALGORITHM

The main idea of the FDE algorithm is to use neighboured spatial/temporal candidates as input for a block-recursive disparity estimation. It is motivated by the assumption that most likely at least one of these candidate vectors will provide a good predictor or even the correct value for disparity at the current pixel-position. Apart from a considerable reduction of computational load this method also leads to spatio-temporally consistent disparity vector fields. The second aspect is that it is particularly important to avoid temporal inconsistencies in disparity sequences, which may cause strongly visible and very annoying artefacts in virtual views synthesised on the basis of these disparities. However, taking into account the case that none of the candidates delivers a suitable vector, a further update vector is tested against the best candidate. This update vector is computed by applying a local pixel-recursive process to the current block. It uses the best candidate of block-recursion as a start vector. As shown in Figure 5, the whole algorithm can be divided into the following three stages:

1. Three candidate vectors (two spatial and one temporal) are evaluated for the current block position by using recursive block matching.
2. The candidate vector with the best result is chosen as the start vector for the pixel-recursive algorithm which yields an update vector.
3. The final vector is obtained by comparing the update vector from the pixel recursive stage with the start vector from the block-recursive one.

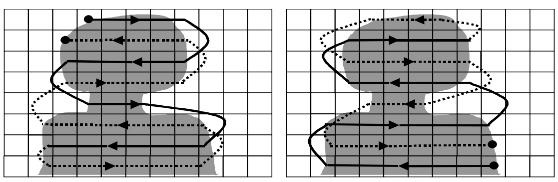


Outline of the fast hybrid recursive disparity estimator
Figure 5

Further details are described in the following sections.

4. BLOCK-RECURSION

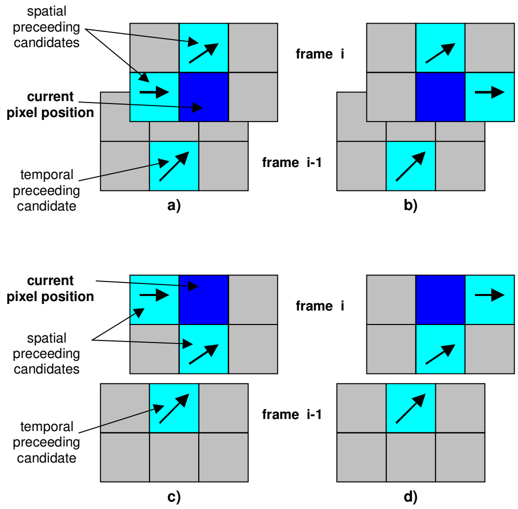
The block recursion is performed in spatial and temporal direction on the grid of a sparse disparity vector field - usually with block sizes of 8x8 or 4x4 pixels. To determine the spatial candidate vectors in the most isotropic way, the video frames are scanned in two interleaved meander paths changing their order from frame to frame. Moreover, as the FDE algorithm is able to cope with arbitrarily shaped video objects (e.g.: the segmented portrayal of a conferee to be integrated seamlessly into a virtual conference room), the interleaved meander is adapted to the binary shape of the video object. In even frames the first run scans the grid in meanders from top to bottom at odd lines and, then, in the second run in meanders from bottom to top at even lines. Vice versa, in odd frames the scan starts with bottom-to-top followed by top-to-bottom. Intensive experiments have proven that this alternating and interleaved meander scan path lead to a better (i.e. faster) convergence of disparity estimation, especially at moving edges where depth discontinuities and occlusions occur. The complete scanning scheme is shown in Figure 6 (solid and dashed scan lines refer to first and second run).



Meander scan for arbitrarily shaped video objects (left: even frames; right: odd frames)
Figure 6

Following this scan path, three candidates are tested to select the best one for the current block-vector position (see Figure 7):

- A vertical predecessor, which is chosen from the block above or below, depending on whether the vertical scan-direction is top-to-bottom or bottom-to-top.
- A horizontal predecessor, which is taken from the left- or the right-neighboured block, depending on the current horizontal scan direction of the meander.
- A temporal predecessor, which is taken from the previous reference frame.



Spatial and temporal candidates for left and right scan directions in the case of a top-to-bottom scan (a and b) and a bottom-to-top scan (c and d)
Figure 7

The three candidates from Figure 7 are compared to find the best match between the current pixel position in the left image and the corresponding pixel position in the right image. The following shape-driven displaced block difference (DBD) using absolute values is taken as criterion for this purpose. For boundary blocks it may happen that some of the three candidate vectors are not available because they are out of the binary mask of the current or previous video object. In addition, some candidate vectors may not be usable because they point to transparent blocks outside the binary mask. In both cases only valid predecessors are used. If all three predecessors are non-valid, the output vector of block-recursion is set to a default value.

$$DBD(\mathbf{d}) = \sum_{x=0}^{M}\sum_{y=0}^{N} s_t(x,y)\cdot\left|f_t(x,y) - f_{t-1}(x+d_x, y+d_y)\right| \qquad (4)$$

with

$$s_t(x,y) = \begin{cases} 1 & ,if \quad (x,y) \;\; inside\ object \\ 0 & ,if \quad (x,y) \;\; outside\ object \end{cases} \qquad (5)$$

Notice that no local search around the candidate vector is applied in the block-recursive stage. Thus, if the block-recursive stage is used alone without any other matching technique, the "best match vector" would always be chosen from the same triple of candidates. This condition is not bad as long as the algorithm works in an area with homogeneous or spatio-temporally consistent disparities. However, it fails as soon as the matcher has to cope with changes or discontinuities in the disparity map. Therefore, to be able to escape from the candidate triple in such a situation, the output of the block-recursive stage ("start vector" in Figure 5) has to be updated permanently. This update of the start vector is delivered by the pixel-recursive stage which is explained in more detail in the next section.

5. PIXEL-RECURSION

Pixel-recursive disparity estimation is a low-complex method to calculate dense displacement fields using the optical-flow principle. Following this principle, the update vector is calculated with respect to spatial gradients in the current frame and the displaced difference given by corresponding points in the left and the right image. The displaced pixel difference (DPD) computation is dependent on an initial displacement vector $\boldsymbol{d}_i$. The updated displacement vector $\boldsymbol{d}$ is obtained as follows:

$$\mathbf{d}(x,y) = \mathbf{d}_i - \varepsilon \cdot DPD(\mathbf{d}_i, x, y) \cdot \frac{\mathbf{grad}\ f(x,y)}{\left\|\mathbf{grad}\ f(x,y)\right\|^2} \qquad (6)$$

where ε describes a so-called convergence factor.

Strictly speaking, eq. (6) has to be performed iteratively until a minimum DPD is reached while using the output of the previous iteration step as initial displacement vector for the next one. However, as the pixel-recursive stage is only used for finding an update vector, the following approximation is applied here

$$\mathbf{d}(x,y) = \mathbf{d}_i - DPD(\mathbf{d}_i, x, y) \cdot \left[u_x, u_y\right]^T \qquad (7)$$

with

$$u_x = \begin{cases} 0 & ,if\ \dfrac{\delta f(x,y)}{\delta x} < \Theta \\ \left[\dfrac{\delta f(x,y)}{\delta x}\right]^{-1} & ,else \end{cases}$$

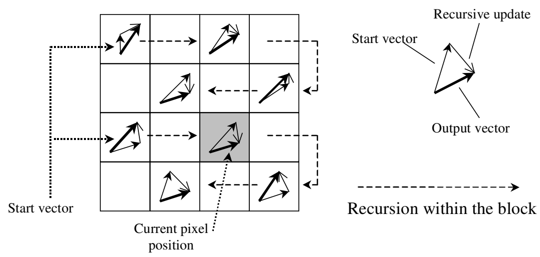
$$u_y = \begin{cases} 0 & ,if\ \dfrac{\delta f(x,y)}{\delta y} < \Theta \\ \left[\dfrac{\delta f(x,y)}{\delta y}\right]^{-1} & ,else \end{cases} \qquad (8)$$

and

$$\frac{\partial f(x,y)}{\partial x} \approx \frac{f(x+1,y) - f(x-1,y)}{2}$$

$$\frac{\partial f(x,y)}{\partial y} \approx \frac{f(x,y+1) - f(x,y-1)}{2} \qquad (9)$$

Experiments have shown that there is no notable difference between the original optical flow relation from eq. (6) and its approximation in eq. (7). The threshold value in eq. (8) is usually set to a value of two or three. It decreases the sensitivity of pixel-recursion to noise in unstructured image regions.



Outline of pixel-recursion scheme
Figure 8

Multiple pixel-recursion processes are started at every first pixel position of the odd lines in the block under inspection (this is shown in Figure 8 for an example of 4x4 block). Each recursion works over two lines using left-to-right scan for odd and right-to-left scan for even lines. Thus, the total number N of recursions per block depends on the size of the block. It is given by its vertical length divided by two (i.e. $N=2$ in the example from Figure 3, $N=4$ for 8x8 block, etc.).

If a rectification process has previously been applied to the original views, pixel recursion is only used for the x-component of the disparity vector in eq. (7), eq. (8) and eq. (9), respectively, because the epipolar lines always coincide with the horizontal lines of the frame (see section 2). However, in the case of arbitrarily oriented epipolar lines (direct method without rectification, see section 2) pixel-recursion is carried out for both components of the disparity vector. Here, the x-and y-components are processed independently from each other and, as a consequence, the resulting update vector does not necessarily meet the epipolar constraint, although the initial vector is exactly at the corresponding epipolar line or at least very close to it. Therefore, the update vector is clamped to the closest pixel position at the current epipolar line after each recursion step.

Subsequently, the vector with the smallest DPD from all pixel-recursion processes is taken as final update vector. After pixel-recursion, the DBD is calculated for this selected update vector and compared to the DBD of the start vector. If the DBD of the update vector is smaller than the one of the start vector, the update vector is chosen as final output vector, otherwise the start vector from the block-recursive stage is retained (see Figure 5).

6. CONSISTENCY & POST-PROCESSING

The previously described matching procedure is performed twice, once for the Left→Right disparity analysis and then for the Right→Left disparity analysis. These two disparity fields allow the use of a very efficient consistency check. In the case of correct disparities the difference between two corresponding vectors should be close to zero. Otherwise, the estimated correspondence is obviously wrong due to occlusions, homogeneous regions or other reasons for mismatches like periodical structures. Therefore, if the difference between the Left→Right and the Right→Left disparity is greater than a predefined threshold, the disparity vector is rejected and has to be interpolated by the surrounding disparity vectors surviving the consistency check.

The resulting holes in the sparse field are firstly filled by a 3x3 median filter. This procedure also smoothes the sparse vector field and filters out outliers. But, obviously, it is not possible to apply the median filter to those holes which are larger than the filter mask itself. Therefore, to fill large holes, a linear interpolation filter is applied in the horizontal direction. Subsequently, a bilinear filter is used to generate a dense disparity vector field out of the sparse field.

7. EXPERIMENTAL RESULTS

Table 1 presents results of measurements on the computation time of the FDE algorithm for one video frame in full ITU-Rec. 601 resolution. The measurements have been carried out with professional profiling tools on a state-of-the-art processor (PIII, 800 MHz). The whole FDE algorithm concerning the sparse disparity vector field including Left→Right and Right→Left matching, consistency check and post-processing have been taken into account during the measurements. The up-conversion from sparse to dense vector fields, however, has been excluded, because this part represents a standard process (i.e. bilinear filtering) which can be implemented more efficiently by specialised processors or dedicated hardware.

A comparison of the two methods under study shows that their computation times are in the same order of magnitudes. In principle, the horizontal matching in rectified images is a little bit faster. This has two reasons. Firstly, the clamping onto epipolar lines at the pixel-recursive stage (see Figure 5) is not needed in this case and, thus, the corresponding computation time can be saved. Secondly, pixel-recursion is only carried out for the x-component of the disparity vector and, as a consequence, the computational amount of pixel-recursion is halved.

Grid size of the sparse field	horizontal matching in rectified images	matching along arbitrary epi-polar lines
4x4	128 ms	145 ms
8x8	35 ms	37 ms

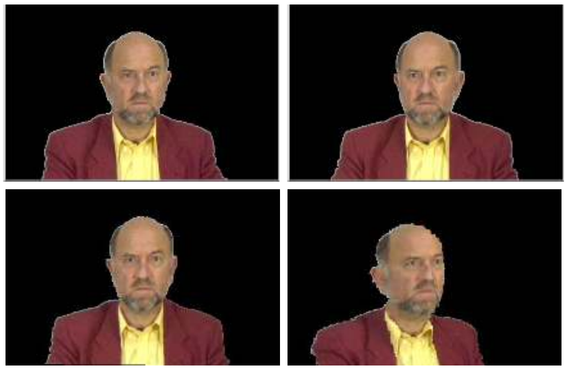
Computation time of FDE algorithm for one ITU-Rec. 601 frame at Pentium III, 800 MHz
Table 1

However, the rectified images are usually much larger than the original ones - at least for camera configurations related to the application scenario sketched in Figure 2. This can also been seen in Figure 4. Notice that the rectified images have been down-scaled here for illustration purposes. Supposing a comparable spatial resolution of the video objects, the rectified images obviously become much larger than the original ones and, therefore, much more pixels have to be processed. Clearly, this increases the computation time and, as it can be seen from the numbers in Table 1, it almost compensates the savings in algorithmic complexity discussed before.

Moreover, the numbers in Table 1 do not include the computation time for the rectification process itself. Related measurements have shown that this pre-process needs 80 to 240 ms per frame at a Pentium III, 800 MHz for video in ITU-Rec. 601 resolution. The exact figure depends on the quality of the interpolation filter used during rectification. Most likely, this amount can be decreased by using special tools for standard warping applications. Nevertheless, summarising all pros and cons, it can be concluded that horizontal matching in rectified images is rather more than less complex compared to direct matching along the arbitrarily oriented epipolar lines, due to the extra processing required by the rectification. Therefore, it was decided to concentrate further work on the direct matching method along arbitrarily oriented epipolar lines.

For real-time processing of video with full ITU-Rec. 601 resolution the computation time for one frame must fall below 40 ms. Thus, following the numbers in Table 1 real-time processing can be achieved for sparse fields with block sizes of 8x8 pixels. For 4x4 pixel resolution, however, a further reduction by a factor of 3 to 4 is has to be achieved. Nevertheless, it can be stated that the algorithm is quite fast and that is suitable for the application under consideration. There is a lot of potential for further run-time optimisations, especially at the pixel-recursive stage. Thus, envisaging further optimisations as well as next processor generations, a real-time implementation of a FDE algorithm with 4x4 block sizes seems to be realistic in near future.

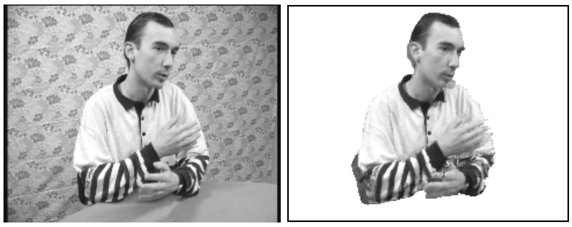
In this context the experiments have also shown that a sparse field resolution of 4x4 pixels is absolutely sufficient to provide a good quality whereas a 8x8 block size represents an acceptable trade-off making real-time processing feasible with today's processor technology. The quality at 4x4 resolution is indeed excellent for typical head-shoulder scenes which represent more or less a convex object with limited amount of occlusions and discontinuities in depth. This is shown in Figure 9 and Figure 10 for the test sequences CLAUDE (weakly convergent set-up) and PUPPET (strongly convergent set-up), respectively.



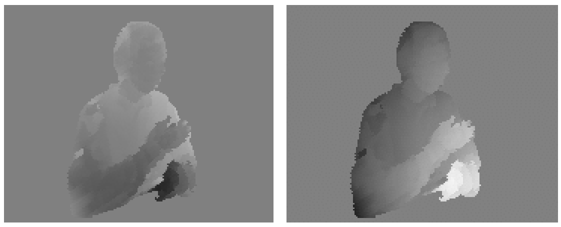
Top: original left and right view of CLAUDE
Bottom left: virtual view at middle of baseline
Bottom right: virtual view from bottom-left
Figure 9

In addition, Figure 11 presents results of a ground truth test, which is more realistic for the immersive tele-conference scenario. The comparison between the reference image captured at the ground truth position and the corresponding virtual view synthesised on the basis of the original views from Fig. 4 (top) show that the FDE algorithm works well in object areas with continuous depth. However, problems occur in regions with occlusions and discontinuities (arms in front of body). This is not surprising at all, because the only part of the FDE algorithm that is able to cope with discontinuities is the pixel-recursive stage and, in fact, it is not very accurate in this sense. Nevertheless, the disparity vector fields from Figure 12 demonstrate that the front layers (i.e. the arms) can at least be detected with high robustness and reliability. Thus, the problem is mainly a question of refinement and accuracy at the segmentation border. Several approaches to solve this refinement problem are currently under study and the first results are promising.

Top: original central and right view of PUPPET
Bottom: virtual views from different directions
Figure 10



Reference and virtual view for ground truth test
Figure 11



Horizontal and vertical component of
sparse disparity vector field (Left→Right)
Figure 12

8. CONCLUSION AND OUTLOOK

The disparity estimator presented in this paper is based on a recently proposed matching algorithm which has originally been developed for fast motion estimation in fields of standards conversion and MPEG coding. To exploit this approach for fast disparity estimation in high quality applications like immersive tele-conferencing, the algorithm has been modified such that it respects the epipolar constraint. Hence, it is able to match point correspondences between strongly convergent stereo images by searching along arbitrarily oriented epipolar lines.

Experiments have shown that pure software implementations of this algorithm process video images with full ITU-Rec. 601 resolution in real-time on state-of-the-art processors (PIII, 800 MHz) and that virtual views can be synthesised with reasonable quality on the basis of the resulting dense disparity vector fields.

The future work will therefore be concentrated on the following two items. The first one is a further improvement of the pixel-recursive stage. Here, it might be very interesting to introduce a λ-parameterisation as it has been proposed in [Alvarez00] in the framework of an anisotropic diffusion approach for disparity estimation. A potential advantage of this method is given by the fact that the pixel-recursion runs only one λ-parameter along the epipolar line instead of the two independent x- and y-components which have then again to be clamped onto the epipolar line. Thus, the intention of such an extension is to stabilise the matching results and to halve the computational amount of pixel-recursion. The second point is an improvement of the dense disparity vector field in areas with depth discontinuities. Here it is foreseen to develop a segmentation driven refinement which detects discontinuities implicitly by exploiting results of the consistency check and by analysing the sparse disparity field. In critical regions the filling of holes in the sparse disparity vector field is then adapted to results of a local and implicit texture segmentation.

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

[Alvarez00] 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.

[Barron94] J. L. Barron, D. J. Fleet and S. S. Beauchemin: „Performance and Optical Flow Techniques“, *Int. Journal on Computer Vision*, Vol. 12, No.1, 1994.

[Bertozzi98] 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.

[Faugeras93] O.D. Faugeras et al: „Real-time Correlation-Based Stereo: Algorithm, Implementations and Applications“, *INRIA Research Report No. 2013*, INRIA, Sophia-Antipolis, August 1993.

[Fusiello97] E. Fusiello, E. Trucco, A. Verri: „Rectification with Unconstrained Stereo Geometry“, *British Machine Vision Conference 1997*, Essex, pp.400-409, September 1997.

[Kauff00] P. Kauff and Klaas Schüür: „A Real-Time A Real-Time MPEG-4 Software Video Encoder Using a Fast Motion Estimator Based on Hybrid Recursive Matching“, *ACM Multimedia 2000*, Los Angeles, October 2000.

[Ohm97] J.-R. Ohm and K. Rümmler: „Variable-Raster Multiresolution Video Processing with Motion Compensation Techniques“, *Proc. of IEEE Int. Conf. on Image Processing*, ICIP-97, Oct. 1997.

[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.

[Robert97] L.Robert, C. Zeller, O.D. Faugeras, M. Hebert: „Applications of Non-Metric Vision to Some Visually-Guided Robotic Tasks“ *in Y. Aloimonos (ed.), Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, pp.-89-134, Lawrence Erlbaum Associates, 1997.

[Schäfer00] R. Schäfer, P. Kauff and O. Schreer: „Tele-Immersion in Shared Presence Conference Systems“, *Proc. of IBC 2000, International Broadcast Convention*, Amsterdam, September 2000.

[Schreer00] O. Schreer, N. Brandenburg and P. Kauff: „A Comparative Study on Disparity Analysis based on Convergent and Rectified Views“, *British Machine Vision Conference 2000*, Bristol, September 2000.

[Smolic98] A. Smolic, K. Rümmler, J.-R. Ohm, R. Schäfer, S. Bauer: „MPEG-4 Video Transmission over DAB/DMB: Joined Optimisation of Encoding and Format Conversion“, *Proc. of 5th Intl. Workshop on Mobile Multimedia Communication,* MoMuc`98, Berlin, October 1998.

[Zhang96] Z. Zhang and G. Xu: „Epipolar Geometry in Stereo, Motion and Object Recognition“, *Kluwer Academic Publisher*, Netherlands, 1996.