REAL-TIME DISPARITY MAPS FOR IMMERSIVE 3-D TELECONFERENCING BY HYBRID RECURSIVE MATCHING AND CENSUS TRANSFORM

E. Trucco[1], K Plakas[1], Nicole Brandenburg[2], Peter Kauff[2], Michael Karl[2], Oliver Schreer[2]

(1) Dept. of Computing and Electrical Engin.
Heriot-Watt University
Edinburgh EH14 4AS
UK
{mtc,costas}@cee.hw.ac.uk

(2) Heinrich-Hertz-Institut
Einsteinufer 37
10587 Berlin
Germany
{kauff, brandenburg, karl, schreer}@hhi.de

Corresponding author: K. Plakas, costas@cee.hw.ac.uk

REAL TIME DISPARITY MAPS FOR IMMERSIVE 3-D TELECONFERENCING BY HYBRID RECURSIVE MATCHING AND CENSUS TRANSFORM

ABSTRACT

This paper presents a novel, real-time disparity algorithm developed for immersive teleconferencing. The algorithm combines the Census transform with a hybrid block- and pixel-recursive matching scheme. Computational effort is minimised by the efficient selection of a small number of candidate vectors, guaranteeing both spatial and temporal consistency of disparities. The latter aspect is crucial for 3-D videoconferencing applications, where novel views of the remote conferees must be synthesised with the correct motion parallax. This application requires video processing at ITU-Rec. 601 resolution. The algorithm generates disparity maps in real time and for both directions (left-to-right and right-to-left) on a Pentium III, 800 MHz processor with good quality.

Keywords
Real-time disparity analysis, teleconferencing, recursive block-matching, pixel-recursive matching, Census.

1. INTRODUCTION

This paper presents a novel, real-time disparity algorithm developed for an immersive teleconferencing system.
Immersive teleconferencing subtends a class of teleconferencing systems enabling conferees located in different geographical places to meet around a virtual table, appearing at each station in such a way to create a convincing impression of *presence* [Kalawsky 00]. In particular, *immersive* indicates a setup which completely covers the angle of view of the visual system such that the boundaries of the station are not visible. Figure 1 shows an artist's rendering of such a system. The purpose is to enable the participants to make use of rich communication modalities as similar as possible to those used in a face-to-face meeting (e.g., gestures, eye contact, realistic images, correct sound direction, etc) and eliminate the limits of non-immersive teleconferencing, which impoverish communication (e.g., face-only images in separate windows, unrealistic avatars, no eye contact) or skew the participants' balance (e.g., some participants appearing larger than others, or in privileged positions on screen. Such a system is the target of the European IST project VIRTUE [Schreer00,VIRTUEweb]. The first demonstrator, currently under development, is limited to semi-immersive displays like 60" plasma monitor or 80" rear Projector.

Figure 1. An artist's rendering of a -immersive teleconferencing setup.

In the teleconferencing system we are developing, the realism of the images of remote conferees is maximised through *view synthesis* [Avidan97]. Two stereo pairs collect images of each conferee in each station; from these, 3-D disparity maps are computed at frame rate, and used to generate synthetic views of remote participants in the remote stations, adapted to the viewpoints of the local participants. In this way, and using physically-plausible view synthesis, the image generated is a *true* image, not an avatar carrying unrealistic artifacts.

A key ingredient in such a system is a module computing reliable disparity maps at frame rate. Requirements are exacting: full resolution video processing according to ITU-Rec. 601; disparity maps in real time; no constraints on participants (e.g., clothes, visual markers); and cameras mounted around a wide screen, yielding a *wide-baseline* stereo geometry, that is, significant image differences caused by large camera displacement and different orientations. Figure 2 shows an example.

Figure 2. Example of stereo pair typical for our application. Notice the large viewpoint difference (wide baseline) between the two images.

These requirements are not met *in toto* by existing approaches, typically based on hierachical block-matching [Faugeras93] or optic flow [Barron94]. Wide-baseline stereo has been investigated [Intille94,Pritchett98] but not necessarily in real-time applicative contexts. On the other hand, several real-time stereo systems have been built around the parallel-camera configuration to minimise computational complexity (see e.g. [Bertozzi97, Faugeras93, Konolige97, Ohm98], [Redert97] for a teleconferencing application, and [Kanade96, Triclops] for other real-time stereo systems).

The algorithm proposed in this paper combines an optimised implementation of *Census-based matching* [Zabih94] with a new, hybrid recursive matching algorithm (HRM). We achieve real-time disparity maps reduced by a factor of 8 with respect to full-size CCIR601. The attraction of Census-based matching is its low cost, as only shifts and integer operations are performed, and improved performance in discontinuity regions.

HRM reaps the advantages of both block-recursive matching and pixel-recursive optical flow estimation. This algorithm has already been used successfully for fast motion estimation for format conversion and MPEG coding [Kauff00,Ohm97] and has recently been applied to disparity estimation [Kauff 01].

This paper is organised as follows: Section 2 reviews briefly the HRM algorithm; Section 3 presents Census-based block matching; Section 4 sketches the issues of consistency and post-processing; Section 5 presents some experimental results; Section 6 summarises and discusses our work, and suggests some future developments.

2. HYBRID RECURSIVE MATCHING

The structure of the HRM algorithm is illustrated in Figure 3. The Census transform and Hamming distance correlation replace SAD in calculating the Displaced Block Difference (DBD) within the block matching stage of the HRM.

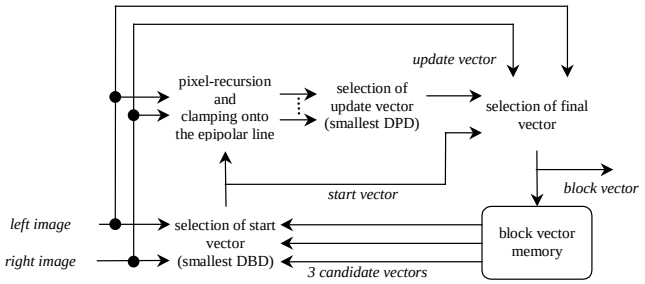

Figure 3: outline of HRM algorithm.

The main idea of HRM is to use neighbouring spatio-temporal candidates as input for block-recursive disparity estimation. The rationale is that such candidate vectors are the most likely to provide a good estimate of the disparity for the current pixel. In addition, 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, which uses the best candidate of block-recursion as a start vector. Apart from a considerable reduction of computational load, this method also leads to spatio-temporally consistent disparity maps, particularly important as temporal inconsistencies in disparity sequences may cause obvious, annoying artifacts in the final virtual images of participants.

The whole algorithm can be divided into three stages (Figure 3):

1. three candidate vectors (two spatial and one temporal) are evaluated for the current block position by 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.

Census matching is used in steps 1 and 3 for block matching, *in lieu* of SAD.

2.1 BLOCK RECURSION

Block recursion is performed in the spatial and temporal directions on the grid of a sparse disparity vector field, usually with 8x8 or 4x4 grid size. To cope with arbitrarily shaped video objects and to determine the spatial candidate vectors isotropically, the video frames are scanned in two interleaved, meandering paths, changing their order from frame to frame and guided by a binary mask representing the shape of the video object [Kauff 01]. Three candidates are tested to select the best one for the current block-vector position (see Figure 4):

- A *vertical predecessor*, 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*, taken from the left or right neighbour block, depending on the current horizontal direction of the scan path.
- A *temporal predecessor*, taken from the previous reference frame.

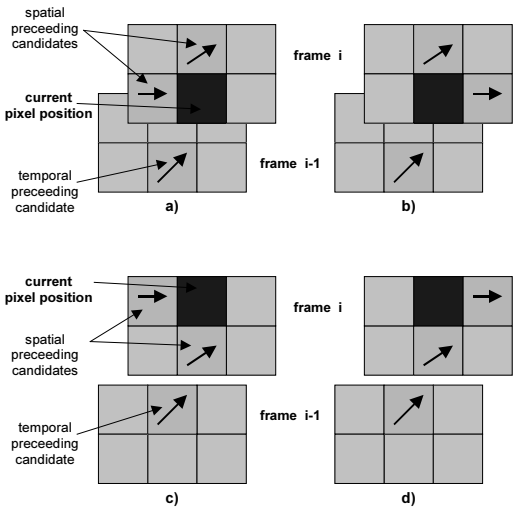

Figure 4: spatial and temporal candidates for left and right scan direction,s in the case of a top-to-bottom scan (a and b) and a bottom-to-top scan (c and d).

The three candidates are compared to find the best match in the right image for the current pixel in the left image. In the HRM original version, the following shape-driven displaced block difference (DBD) was taken as criterion for this purpose, and has been replaced by Census:

$$DBD(\mathbf{d}) = \sum_{x=0}^{M}\sum_{y=0}^{N} s_l(x,y)\cdot\left|f_l(x,y) - f_r(x+d_x, y+d_y)\right|$$

where

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

Notice that no local search around the candidate vector is applied in the block-recursive stage. Thus, if only block-recursive matching were used, the "best match vector" would always be chosen from the same triple of candidates. This works well where disparities are spatially and temporally consistent, but fails in the presence of abrupt changes in the disparity map. To obviate this problem, the output of the block-recursive stage must be updated permanently. The update is delivered by the pixel-recursive stage, explained in the next section.

2.2 PIXEL RECURSION

Pixel-recursive disparity estimation is a low-complexity method calculating dense displacement fields using a simplified optical flow approach. The update vector, $\boldsymbol{d}$, is calculated using spatial gradients in the current frame and the displaced pixel difference DPD given by corresponding points in the left and right images as follows:

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

$$with \quad DPD(\mathbf{d}_i, x, y) = \left|f_l(x,y) - f_r(x+d_x, y+d_y)\right| \qquad (1)$$

where ε describes a so-called convergence factor and $\boldsymbol{d}_i$ is an initial displacement. Strictly speaking, Eq. (1) must be iterated until a minimum DPD is reached, setting $\boldsymbol{d}_i$ to the output of the previous iteration. However, as the pixel-recursive stage is only used for

finding an update vector, the following approximation is applied:

$$\mathbf{d}(x,y) = \mathbf{d}_i - DPD(\mathbf{d}_i, x, y) \cdot \left[u_x, u_y\right]^T \qquad (2)$$

with

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

$$u_y = \begin{cases} 0 & , if \ \dfrac{\delta f(x,y)}{\delta y} < \Theta \\ \left[\dfrac{\delta f(x,y)}{\delta y}\right]^{-1} & otherwise \end{cases} \qquad (3)$$

and

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

$$\frac{\partial f(x,y)}{\partial y} \approx \frac{f(x,y+1) - f(x,y-1)}{2} \qquad (4)$$

Experiments have shown that there is no notable difference between the original optic flow relation in Eq. (1) and its approximation in Eq. (2). The threshold value in Eq. (3) is usually set to 2 or 3, and decreases the sensitivity of the pixel recursion to noise in unstructured image regions.

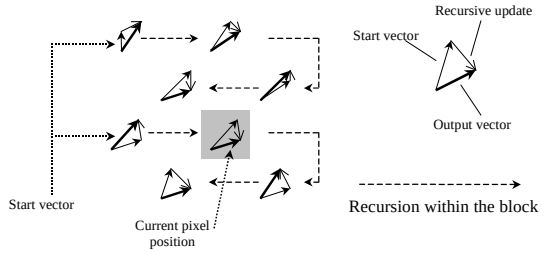

Figure 5: outline of the pixel-recursion scheme.

Multiple pixel-recursive processes are started at every first pixel position of the odd lines in the block under inspection (an example is shown in Figure 5 with a 4x4 block). Each recursion works over two lines using left-to-right scan for odd lines and right-to-left scan for even lines. Thus, the total number *N* of recursions per block depends on the size of the block, and is given by half of the block's height (i.e., *N=2* in the example from Figure 5, *N=4* for 8x8 blocks, etc).
With rectified images, pixel recursion is only used for the x-component of the disparity vector in Eqs. (2), (3) and (4). With unrectified images, pixel recursion is carried out for both components of the disparity vector. Here, the x and y components are processed independently of each other; as a consequence, the resulting update vector does not necessarily meet the epipolar constraint. Therefore, the update vector is clamped to the closest pixel position at the current epipolar line after each recursion step.
Finally, the vector with the smallest DPD among all pixel-recursion processes is taken as the 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 latter 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 3).

2.3 EPIPOLAR CONSTRAINT

As the HRM algorithm from Figure 1 is used to estimate disparities and since it can therefore be assumed that the cameras of the stereo rig are weakly calibrated, the well-known epipolar constraint can be exploited to increase matching robustness in the given approach. The epipolar constraint tells us that a pixel $\mathbf{m_r}=[x_r,y_r,1]^T$ of the right image which corresponds to a pixel $\mathbf{m_l}=[x_l,y_l,1]^T$ of the left image, must lie on the epipolar line $\mathbf{l_r}$ of $\mathbf{m_l}$ [Zhang 96]:

$$\mathbf{m}_r^T \cdot \mathbf{l}_r = 0 \quad with \quad \mathbf{l}_r = \mathbf{F} \cdot \mathbf{m}_l \qquad (5)$$

Here, **F** denotes the the fundamental matrix representing a compact description of weak camera calibration.

However, note that in the most general case the two components of the gradient vector in Eq. (3) are calculated independently from each other and do not necessarily meet the epipolar constraint. One way to correct this is to clamp all vectors obtained during pixel recursion onto the closest position of the corresponding epipolar line. Thus, the update vectors and with it all output vectors subsequently stored in the block vector memory and used as candidates during block recursion (see Fig. 3) now respect the epipolar constraint.

Another possibility is to rectify the left and right input images before applying the HRM

algorithm from Figure 3. In this case the epipolar lines always coincide with horizontal scan lines of the rectified images [Fusiello97]. As a consequence, disparity estimation is simplified to a horizontal match. In this special case, only the horizontal component of the gradient in Eq. (3) is calculated and only the horizontal component in Eq. (2) is updated whereas the vertical component is always equal to zero. The epipolar constraint is now respected implicitly.

A further alternative to exploit the epipolar constraint implicitly is the so-called λ-parametrisation which has been proposed by [Alvarez00] in the framework of an anisotropic diffusion approach to disparity estimation. It is based on the fact that the disaprity vector can be decomposed into independent components

$$\mathbf{m}_r = \mathbf{m}_l + \mathbf{d} = \mathbf{m}_l + \gamma \cdot \mathbf{N} + \lambda \cdot \mathbf{T} \qquad (6)$$

Here, **N** and **T** denote normalized vectors perpendicular and, respectively, tangential to the corresponding epipolar line $\mathbf{l_r}$. γ is the distance from $\mathbf{m_l}$ to $\mathbf{l_r}$ and λ is an unknown parameter along the epipolar line. Note that **N**, **T** and γ are given by epiolar geometry and can be calculated in dependence on $\mathbf{m_l}$ and $\mathbf{l_r}$. Using this λ-parametrisation the pixel recursion from Eq. (2) can be applied to the scalar λ instead of disparity vector **d**:

$$\lambda = \lambda_i - DPD(\mathbf{d}_i, x, y) \cdot [w_x \; w_y] \cdot [u_x \; u_y]^T \qquad (7)$$

Here, w_x and w_y denote to weighting factors depending on $\mathbf{m_l}$ and $\mathbf{l_r}$. After pixel recursion the resulting update vector can be determined using Eq. (6).

3. CENSUS-BASED MATCHING

The original HRM algorithm from Fig. 3 uses the DBD to select the start vector out of the three candidates. To combine HRM with census-based matching, the algorithm has been modified as shown in Fig. 6. For this purpose the two input images are transformed by using the *Census Transform*, and the matching criterion adopted is the Hamming distance (HD).

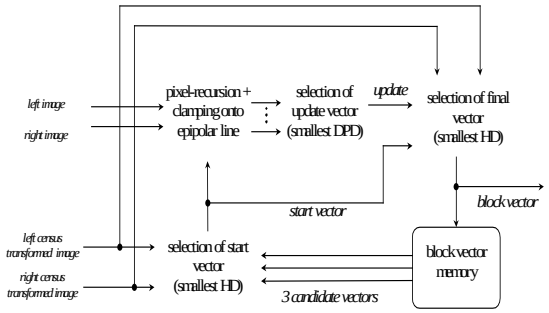

Figure 6: combination of HRM with CENSUS.

The *Census transform* [Zabih94] captures the local grey-level structure. It maps the neighbourhood of a pixel to a bit string identifying the pixels with smaller intensity values than the central one. As no intensity value is encoded, the representation is largely immune to photometric and projective distortion, image gain or bias (whether local or global) and gamma correction.
Assume that the image region to transform is a square of side β, and W is the square correlation (matching) window. Each pixel P in W will be replaced with the value of the local transform, computed in the region of radius β centered at P. Note that the transform is applied to W separately in I_1 and I_2, then the transformed windows are compared via correlation.
If $\oplus$ denotes concatenation, the Census transform of a pixel P of image I is defined as

$$T[I,P] = \bigoplus_{P' \in W(P,\beta)} \xi(I,P,P'),$$

where

$$\xi(I,P,P') = \begin{cases} 1 \text{ if } I(P) < I(P') \\ 0 \text{ otherwise} \end{cases}$$

and $I(P)$ and $I(P')$ are, respectively, the intensity values at points P and P'. The correlation of the window W between the two images I_1 and I_2 is given by

$$C(W) = \sum_{P \in W} \sum_{P' \in W'(P,\beta)} \left| Err(P,P') \right|$$

where the error (dissimilarity) criterion is

$$Err(P,P') = \xi(I_1,P,P') - \xi(I_2,P,P')$$

Note that the value of *Err* can be 0, 1 or -1, and that $Err(P,P')$ is nonzero just in case P and P'

switch their relative ordering between the two images. If $\otimes$ denotes the Hamming distance between two bit strings, i.e., the number of bits that differ, the Census transform correlation can be written as

$$C(W) = T[I_1, P] \otimes T[I_2, P].$$

Both steps of this algorithm (Census transform, correlation) are spatially uniform. However, when combined, the resulting method is not: the influence of a particular pixel varies depending on its location, and the farther a pixel is from the center of the window, the lower its influence.

The attraction of Census-based matching is its low cost, as only shifts and integer operations are performed. Our implementation [Zini99] adopts various optimisations to minimise the number of operations performed.

4. CONSISTENCY CHECK

The matching procedure described above is performed twice (left-to-right and right-to-left), then left-right consistency is applied. Disparity vectors failing the test are rejected and are interpolated using the surrounding, consistent disparity vectors.

The holes created by this consistency analysis are firstly filled by a 3x3 median filter. This procedure also smooths the disparity map and filters out outliers. Obviously, the median filter cannot be applied to holes larger than the filter mask. To fill these, 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.

5. EXPERIMENTAL RESULTS

The benefits of the Census transform within the HRM framework have been tested on the basis of a segmented and rectified input sequence, of which a stereo pair is shown in Figure 7.

Figure 7: input stereo pair after figure-background segmentation and rectification.

Figure 8 shows the disparity maps obtained from the combined HRM and Census algorithm (Figure 6). The black regions show areas of inconsistent disparities.

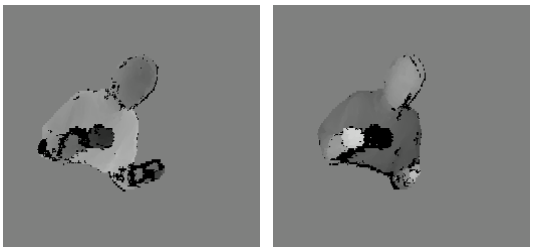

Figure 8: left-to-right and right-to-left disparity maps after consistency check.

Figure 9 shows the same disparity maps after median filtering and interpolation. The holes c\aused by the inconsistency check are now filled. Note that the disparity maps are spatially (and temporally) consistent in areas of homogeneous depth, and rapid depth transitions around the arms have been found accurately. Synthesis results based on this disparity maps show good quality.

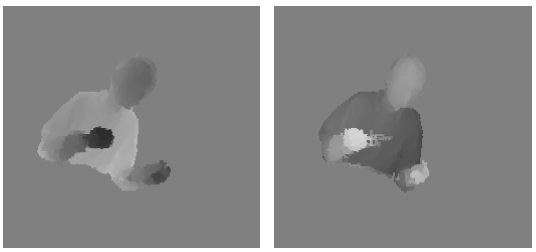

Figure 10: Final disparity maps after median filtering and interpolation.

To quantify the benefit of Census Transform in the given framework, we compared the HRM with and without CT on the basis of computer simulations. Two criteria have been used for this comparison: the *percentage of consistent*

disparities and the *delta of the consistency check*. Figures 11 and 12 plot the two criteria, respectively, against frame number. The grey circles refer to the original HRM algorithm (called SAD here), the black squares to the Census matching version. Both graphs show that CENSUS matching improves performance. More detailed comparisons (not reported here) also show that the combination of HRM with Census matching gives better results especially in critical regions, such as areas of low texture or close to borders of segmented objects (e.g., arm contours in Figure 7).

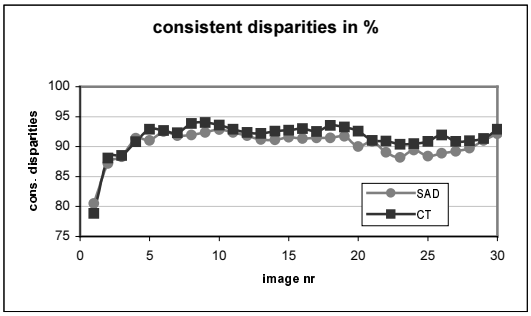

Figure 11: Percentage of consistent disparities (see text).

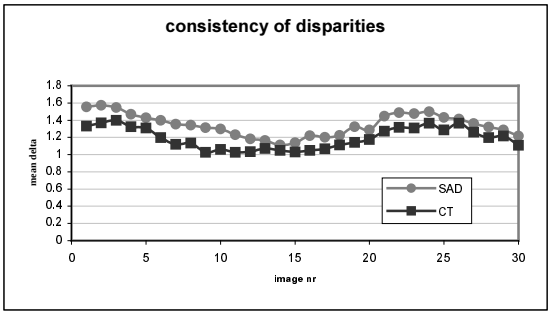

Figure 12: Mean delta of consistency check (see text).

6. CONCLUSIONS AND FUTURE WORK

We have presented a real-time matching algorithm combining Census matching with a highly efficient pixel-recursive scheme. Experiments with stereo sequences from the target application (immersive teleconferencing) indicate that the combined algorithm performs better than the original pixel-recursive one. The algorithm is being incorporated in the first demonstrator of an immersive teleconferencing system developed for by EU VIRTUE project.

As to future technical work, we are investigating a different kind of parametrisation for estimating the epipolar geometry, called *lambda parametrisation*. Experiments are also under way with a novel, robust detector of occlusion contours using registered disparity maps and intensity data.

ACKNOWLEDGMENTS

This work is partially supported by the Ministry of Science and Technology of the Federal Republic of Germany, Grant-No.01 AK 022, and by the EU project VIRTUE (Framework V, IST). Thanks to Emile Hendriks, Francesco Isgro‘ and Bang-Jun Lei for many useful discussions.

REFERENCES

[Avidan97] S. Avidan and A. Shashua: *Novel view synthesis in tensor space*, Proc. IEEE Conf on Comp. Vis. and Patt. Rec., 1997, pp.1034-1040.

[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.

[Barrow94] J. L. Barron, D. J. Fleet and S. S. Beauchemin. Performance of Optical Flow Techniques, *Int. Journal of Computer Vision*, Vol. 12, No.1, 1994.

[Bertozzi98] M. Bertozzi and A. Broggi. GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection, *IEEE Trans. on Image Processing*, Vol.7, No.1, January 1998.

[Intille94] S Intille and A Bobick, *Disparity-Space Images and Large Occlusion Stereo*, Proc European Conf on Computer Vision, Stockholm, 1994.

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

[Fusiello97] E. Fusiello, E. Trucco, A. Verri: *Rectification with Unconstrained Stereo Geometry*, Proc. British Machine Vision Conference, Colchester (UK), 1997, pp.400-409.

[Kalawsky 00] R. Kalawsky: *The Validity of Presence as a Reliable Human Performance Metric in Immersive Environments*, Proc. Presence 2000, 3rd International Workshop on Presence, 2000, Delft, The Netherlands.

[Kanade96] T Kanade, A Yoshida, K Oda, H Kano, M Tanaka: *A Stereo Machine for Video-Rate Dense Depth Mapping and Its New Applications*, IEEE Int Conf on Computer Vision and Pattern Recognition, San Francisco (CA), 1996.

[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.

[Kauff 01] P. Kauff, N Brandenburg, M. Karl and O. Schreer, *Fast Hybrid Block- and Pixel-Recursive Disparity Analysis for Real-Time Applications in Immersive Tele-conference Scenarios*, Proc. of WSCG 2000, 9th Int. Conf. in Central Europe on Computer Graphics, Visualization and Computer Vision, Plzen, Czech Republic, February 2001.

[Konolige97] K Konolige: *Small Vision Systems: Hardware and Implementation*, 8th Int Symp on Robotics Research, Hayama, Japan, 1997.

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

[Ohm98] J.-R. Ohm et al: A real-time hardware system for stereoscopic video conferencing with viewpoint adaptation, *Image Communication*, special issue on 3-D TV, January 1998.

[Pritchett98] P Pritchett and A Zisserman, *Wide Baseline Stereo Matching*, Proc Int Conf on Computer Vision, Bombay, 1998.

[Redert97] P.A. Redert and E.A. Hendriks, *Disparity map coding for 3D teleconferencing applications*, Proc. SPIE VCIP, Vol 3024, San Jose (CA), April 1997, pp. 369-379.

[Schreer00] O. Schreer and P. Sheppard: *VIRTUE - The Step Towards Immersive Tele-Presence in Virtual Video Conference Systems*, Proc. of Works 2000, Madrid, September 2000.

[Triclops] www.pointgrey.com

[VIRTUEweb] http://www3.btwebworld.com/virtue/

[Zabih94] R. Zabih and J. Woodfill: *Non-parametric local transform for computing visual correspondence*. In J.-O. Eklundh, editor, Proc. ECCV94, European Conference on Computer Vision, Stockholm, Sweden, 1994.

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

[Zini99] D. Zini. *Census transform correlation algorithm: an optimized implementation*. Ocean System Laboratory Technical report, Heriot-Watt University, 1999.