January 2001

Compression of Stereo Disparity Streams Using Wavelets and Optical Flow

Thomas Bülow
University of Pennsylvania

Jane Mulligan
University of Pennsylvania

Geraud de Bonnafos
Ecole Polytechnique

Alexandre Chibane
Ecole Polytechnique

Kostas Daniilidis
University of Pennsylvania, kostas@cis.upenn.edu

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-01-36.

Compression of Stereo Disparity Streams Using Wavelets and Optical Flow

Abstract
Recent advances in computing have enabled fast reconstructions of dynamic scenes from multiple images. However, the efficient coding of changing 3D-data has hardly been addressed. Progressive geometric compression and streaming are based on static data sets which are mostly artificial or obtained from accurate range sensors. In this paper, we present a system for efficient coding of 3D-data which are given in forms of 2 + 1/2 disparity maps. Disparity maps are spatially coded using wavelets and temporally predicted by computing flow. The resulted representation of a 3D-stream consists then of spatial wavelet coefficients, optical flow vectors, and disparity differences between predicted and incoming image. The approach has also very useful by-products: disparity predictions can significantly reduce the disparity search range and if appropriately modeled increase the accuracy of depth estimation.

Comments
University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-01-36.

Compression of Stereo Disparity Streams Using Wavelets and Optical Flow

Thomas Bülow[1], Jane Mulligan[1], Geraud de Bonnafos[2], Alexandre Chibane[2], and Kostas Daniilidis[1]

[1] GRASP Laboratory
Department of Computer and Information Science
University of Pennsylvania, Philadelphia, USA
{thomasbl, janem, kostas}@grasp.cis.upenn.edu
[2] Ecole Polytechnique
Paris, Frace

U of Penn, CIS Dept. Technical Report: MS-CIS-01-36

Abstract

Recent advances in computing have enabled fast reconstructions of dynamic scenes from multiple images. However, the efficient coding of changing 3D-data has hardly been addressed. Progressive geometric compression and streaming are based on static data sets which are mostly artificial or obtained from accurate range sensors. In this paper, we present a system for efficient coding of 3D-data which are given in forms of 2+1/2 disparity maps. Disparity maps are spatially coded using wavelets and temporally predicted by computing flow. The resulted representation of a 3D-stream consists then of spatial wavelet coefficients, optical flow vectors, and disparity differences between predicted and incoming image. The approach has also very useful by-products: disparity predictions can significantly reduce the disparity search range and if appropriately modeled increase the accuracy of depth estimation.

This work was partly supported by the German Research Association (Deutsche Forschungsgemeinschaft – DFG) under the grant Bu 1259/2-1.

1. Introduction

The dramatic increase in bandwidth through optical communication motivated many new applications that were hindered by the communications bottleneck. Many of them require the real-time streaming of sensorial data acquired mainly from audio or video modalities. In this paper, we push the envelope in multimedia communications by studying the problem of compression of 3D dynamic streams of real imagery. This problem arises in applications like tele-immersion [7] where a depth-map is online acquired at one site and transmitted to a remote display station where a user can be immersed in the transmitted scene.

This is different than progressively transmitting static graphics data which rather corresponds to the progressive GIF or JPEG transmission. Geometric compression is an ongoing topic in computer graphics with many recent approaches [6, 5]. Our work is related to compression of stereo-sequences [13, 15]. However, the purpose of these approaches is the best reconstruction of the original two images and not of a depth map. From the perceptual point of view we are interested in the minimal loss in depth or disparity accuracy.

In this paper, we assume that a depth map has been acquired by solving the correspondence problem in stereo from multiple views. We assume that the scene is changing and that such a depth map is produced in a constant frame-rate. Coding a 2+1/2 map like a depth map is also different than coding arbitrary surfaces because we have a very simple topology. Dynamic surface compression is actually in our ongoing agenda.

We further assume that in addition to the depth maps we have a texture image fully calibrated with respect to the depth map. In this paper we propose a disparity stream compression scheme that is based on image-based flow estimation and disparity prediction. This is different than approaches which compute the 3D flow of a scene [16] and different than model-based approaches.

We propose a formula for the prediction of the stereo disparity based on the image velocity and the local depth gradient.

Predicting disparity is of interest for at least two purposes:

1. Efficient coding of the sequence of disparity maps.
2. Reducing the disparity computation by restricting the search-space according to the predicted disparity.

We will not study here the second point. Our compression scheme is based on an initial segmentation of the scene in depth-coherent regions. Then, optical flow (either constant or affine model) is computed for each region. Using the optical flow information and the prediction formula we warp the current map to a predicted disparity map. We compute then the difference between the predicted and the actual disparity map. We compress this difference using JPEG-2000. Our coding consists then of the coded difference plus the region descriptions and their flow.

A simple example of the required bandwidth for a single disparity stream (not counting the texture) is 30 fps x (320 x 240) pix/f x 16bits/pix = 37 Mbps. If the JPEG-2000 difference compression for the same resolution yields a compression rate of approximately 60:1. We will show in the results how this example is related to real disparity sequences in a particular application.

The sequel of this paper is structured as follows: In section 2 we first introduce the basic notations and review the relation between disparity and depth information. We then present a differential method for predicting disparity from the preceding disparity map and the optical flow of the two image sequences. Results are presented in section 3 before we conclude and give an outlook on future work in section 4.

2. Predicting Disparity

We begin by briefly describing the relation between disparity and depth information. Assume a scene viewed by a pair of parallel, strongly calibrated cameras with focal length f and baseline length b. A scene point $\boldsymbol{X} = (X, Y, Z)^T$ is projected to $\boldsymbol{x}_L = (x_L, y_L)^T$ in the left image and to $\boldsymbol{x}_R = (x_R, y_R)^T$ in the right image. The origins of the image coordinate systems are at the centers of the respective images. The origin of the world-coordinate systems is the midpoint of the two centers of projection. The X-, x_L- and x_R-axes are parallel to the baseline, connecting the two centers of projection. The Z-axis is parallel to the line of sight of the two cameras. The *disparity* at $\boldsymbol{x}_L$ is defined as $d = x_L - x_R$. Given $\boldsymbol{X}$, f, and b we find $\boldsymbol{x}_L$ and $\boldsymbol{x}_R$ as

$$(x_L, y_L) = f/Z\,(X + b/2, Y) \tag{1}$$

$$(x_R, y_R) = f/Z\,(X - b/2, Y)\,. \tag{2}$$

Solving for Z yields

$$Z = \frac{fb}{d}, \tag{3}$$

i.e. the depth Z of the scene is proportional to the reciprocal disparity.

2.1. The DCCE

Video coding standards like MPEG use motion-compensated prediction in order to exploit temporal redundancies between subsequent frames of image sequences. A motion vector is stored together with each image block, relating image blocks in subsequent frames. (See [8] for a description of the MPEG video compression standard.) In the encoding stage these motion vectors have to be calculated which can be accomplished by estimating the optical flow of the image sequence.

Optical flow estimation relies on the so-called *brightness change constraint equation* (BCCE). Let $I(\boldsymbol{x}, t)$ be the image intensity at point $\boldsymbol{x}$ at time t. The BCCE expresses the assumption that there is no overall change of the image intensity:

$$\frac{d}{dt} I(\boldsymbol{x}, t) = \nabla_{\boldsymbol{x}} I(\boldsymbol{x}, t) \cdot \boldsymbol{v} + \frac{\partial}{\partial t} I(\boldsymbol{x}, t) = 0. \tag{4}$$

Here $\nabla_{\boldsymbol{x}}$ is the spatial Nabla-Operator $\nabla_{\boldsymbol{x}} = (\partial/\partial x, \partial/\partial y)^T$ and $\boldsymbol{v} = (dx/dt, dy/dt)^T$ is the optical flow. See [1] and [2] for an overview of the vast amount of literature on optical flow estimation.

A straightforward extension of this method to disparity images would lead to the introduction of a *disparity flow* which could be estimated by the *disparity change constraint equation* (DCCE). The DCCE is given by

$$\frac{d}{dt} d(\boldsymbol{x}, t) = \nabla_{\boldsymbol{x}} d(\boldsymbol{x}, t) \cdot \boldsymbol{v} + \frac{\partial}{\partial t} d(\boldsymbol{x}, t) = 0, \tag{5}$$

where the disparity image is just treated as a gray-value image. Assume now that the two parallel cameras are viewing a textured (in order to make disparity estimation feasible), fronto-parallel plane. In this case the disparity is constant within the whole field of view which leads to $\nabla_{\boldsymbol{x}} d(\boldsymbol{x}, t) = 0$ and therefore $\frac{\partial}{\partial t} d(\boldsymbol{x}, t) = 0$. The latter is obviously only true if the motion of the plane is purely parallel to the image plane. This restriction to motions with constant distance to the cameras is much too strong for most practical applications where we want to deal with objects approaching the cameras or departing from them. We thus have to abandon the DCCE and look for an alternative equation.

2.2. Differential Disparity Prediction

Instead of treating disparity in the same way as an intensity image we make use of the defining relation between the disparity and corresponding points in the left and right image. We consider the camera pair viewing a surface moving and deforming in time. Let the surface be given in the parametric form

$$\boldsymbol{X} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^3, \quad (\boldsymbol{p}, t) \mapsto \boldsymbol{X}(\boldsymbol{p}, t) \tag{6}$$

The parameter $\boldsymbol{p}$ allows us to identify and track points on the surface over time. A surface point defined by $\boldsymbol{p}$ and t is mapped to the image point with image coordinates $\boldsymbol{x}_L(\boldsymbol{p}, t)$ in the left image and to $\boldsymbol{x}_R(\boldsymbol{p}, t)$ in the right image. We assume that the mapping between the image coordinates and the surface parameters is invertible at any given point in time. Thus, we can express the disparity as follows:

$$d(\boldsymbol{x}_L, t) = \tilde{d}(\boldsymbol{p}(\boldsymbol{x}_L), t) = x_L(\boldsymbol{p}, t) - x_R(\boldsymbol{p}, t). \tag{7}$$

The time derivative of the disparity

$$\frac{d}{dt}\tilde{d}(\boldsymbol{p}, t) = \frac{d}{dt}x_L(\boldsymbol{p}, t) - \frac{d}{dt}x_R(\boldsymbol{p}, t) \tag{8}$$

allows us to predict the disparity at time $t + \delta t$ using a linear Taylor approximation:

$$\tilde{d}(\boldsymbol{p}, t + \delta t) \approx \tilde{d}(\boldsymbol{p}, t) + \delta t \left[\frac{d}{dt}x_L(\boldsymbol{p}, t) - \frac{d}{dt}x_R(\boldsymbol{p}, t)\right]. \tag{9}$$

Here $d/dt\, x_{L\,(R)}(\boldsymbol{p}, t)$ is the velocity of the projection of the 3D surface point $\boldsymbol{X}(\boldsymbol{p}, t)$ to the left (right) image. We will approximate this term by the optical flow estimated from the left and right image sequences. Note, that the predicted disparity $\tilde{d}(\boldsymbol{p}, t + \delta t)$ is given for the next instance in time but for the same value of the surface parameter $\boldsymbol{p}$. Thus, the predicted disparity is given at a new location in terms of image coordinates:

$$\boldsymbol{x}_L(\boldsymbol{p}, t + \delta t) \approx \boldsymbol{x}_L(\boldsymbol{p}, t) + \underbrace{\delta t \frac{d}{dt}\boldsymbol{x}_L(\boldsymbol{p}, t)}_{\delta \boldsymbol{x}_L(\boldsymbol{p}, t)} \tag{10}$$

Using the gradient of the disparity map we can approximate the disparity locally by a linear expansion. This allows to estimate the predicted disparity at the old image coordinates from the predicted disparity at the new image coordinates:

$$d(\boldsymbol{x}_L, t) \approx d(\boldsymbol{x}_L + \delta \boldsymbol{x}_L, t) - \delta \boldsymbol{x}_L \cdot \nabla d(\boldsymbol{x}_L, t). \tag{11}$$

2.3. Region-based Disparity Propagation

Another way to predict disparity in future frames is to exploit the scene structure described in the disparity map itself. A rough segmentation of the disparity image allows us to estimate optical flow values for a sparse set of regions that actually reflect coherent scene components. Points within regions are then uniformly propagated to predict the next disparity frame.

Segmentation is achieved using flood fill or seed fill [14, pp. 137-141], a simple polygon filling algorithm from computer graphics. We want to construct regions of similar disparity so we limit the maximum disparity range. The flood fill spreads until a threshold on the maximum change in disparity from the value at the original seed pixel is violated. This essentially divides the underlying surfaces into patches where depth is nearly constant. Only rectangular image windows are maintained, rather than a convex hull or more complicated structure, as a result we allow regions to overlap. Small regions are attributed to noise and deleted. Nearby or overlapping windows are merged when the difference between the region mean disparities is small. As regions are constructed and merged, a region label is maintained for each valid disparity pixel (see Fig 1). To propagate disparity, a single flow (u_{iL}, v_{iL}) is computed for

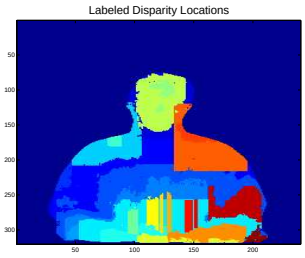

Fig. 1. The disparity values labeled according to the assigned regions.

the entire region R_i for the left image sequence using the familiar Lucas and Kanade method [9]. Region positions are propagated to the right sequence using their mean disparity ($x_R = x_L - \bar{d}_i$), and right flow (u_{iR}, v_{iR}) is calculated. Finally the disparity map $D(\boldsymbol{x}, t)$ is propagated by δt for each pixel $p = (x, y)$ labeled as belonging to extracted region R_i as follows:

$$d(\boldsymbol{x}_L + \boldsymbol{u}_{iL}\delta t, t_1) \approx d(\boldsymbol{x}_L, t_0) + (u_{iL} - u_{iR})\delta t.$$

This approximation is similar to (9) with the only difference that only one flow vector is assigned to each region.

3. Results

In our experiments we use a configuration of three strongly calibrated non-parallel cameras. A correlation technique similar to the one presented in [12] is used for estimating the disparity between the center and right camera. Before estimating the disparities the images are rectified in order to obtain parallel epipolar lines. The disparity value with the highest sum of the correlations in the left and right camera pair is chosen as disparity estimate.

Figure 2 shows one frame of a sequence taken by the three cameras. The resulting disparity map and a side-view of the 3D reconstruction can be seen in Fig. 3, respectively. The reconstruction consists of a set of points. No triangulation is applied to the data.

3.1. Compression of a Single Disparity Frame

The disparity is estimated from three views taken with a rigid setup of three strongly calibrated, non-parallel cameras. The used algorithm is based on the one presented in [12].

Fig. 2. Three views which are used for the 3D reconstruction.

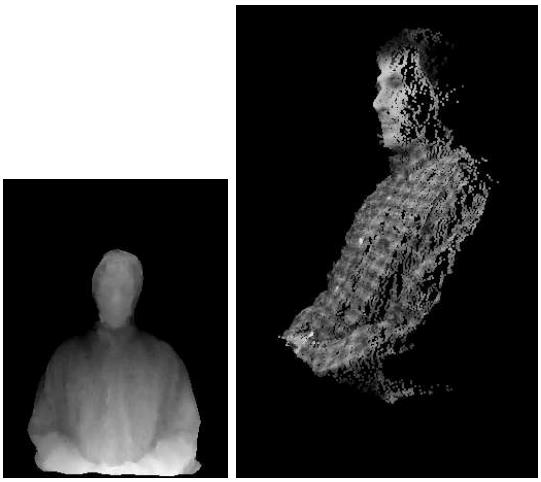

Fig. 3. A disparity map (left) and the 3D reconstruction obtained from it.

Wavelet transform coding schemes have proven to be very efficient in image compression [4, 10]. In order to compress a single disparity frame we use a JPEG 2000 codec which incorporates wavelet based compression[1]. For an overview of the JPEG 2000 standard see [11]. In the current implementation the disparity map is only defined within a foreground mask. Disparity values for the background are not evaluated and are thus not defined. In order to apply the encoder to the disparity map, we assign values to the pixels with undefined disparity. In our experiments we fill the background smoothly from the boundary of the foreground mask to the image boundary. The resulting disparity map is shown in Fig. 5 (right). After the reconstruction of the disparity map from the wavelet coefficients these values are removed. For this reason the foreground mask is stored as well.

Another possibility is to pad the undefined regions with zeros. However, using the zero padded disparity map shown in Fig. 5 (left) leads to distortions at the transition between foreground and background at high compression ratios.

Compression results are shown in Fig. 4. We give the data rate in bits per pixel (bbp), where all the pixels of the disparity map (320 x 240 in our experiments) are counted, whether a value is assigned or not. We also give the number of bits per vertex (bpv) of the final reconstruction. It is worth noting that

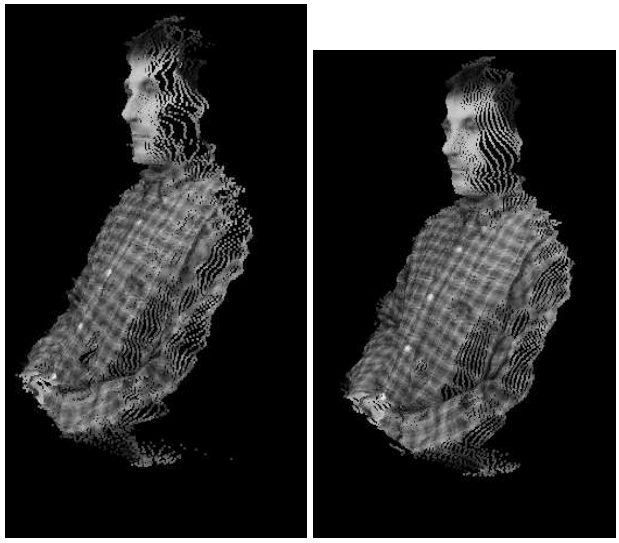

Fig. 4. Reconstruction from compressed disparity map using JPEG 2000 at 0.135 bpp or 0.37 bpv (left) and 0.073 bpp or 0.2 bpv (right).

the reconstructions obtained from highly compressed disparity maps are not only degraded to a certain degree. At the same time they are smoother and outlier are removed to a certain extend which is a natural consequence of discarding high frequency wavelet coefficients.

3.2. Disparity Prediction

In our experiments we compare four methods for the prediction of the disparity at time $t_1 = t_0 + \delta t$ from the disparity at time t_0:

[1] The free demo version of Image Power Inc.'s JasPer 0.072 codec was used.

Fig. 5. A disparity map with the background set to a constant value (left) and with a smooth transition to the background. The values within the foreground mask are the same in both cases.

1. The predicted disparity $d_{P1}(\boldsymbol{x}_L, t)$ at time t_1 is the same as the measured disparity at time t_0.
$$d_{P1}(\boldsymbol{x}_L, t_1) = d(\boldsymbol{x}_L, t_0)$$
2. The predicted disparity $d_{P2}(\boldsymbol{x}_L, t_1)$ is obtained based on the optical flow as in (9). Since this prediction is displaced by the motion vector field between the left frame at time t_0 and t_1, we warp the prediction to the correct position according to the optical flow.
3. The prediction $d_{P3}(\boldsymbol{x}_L, t_1)$ is obtained similarly to $d_{P2}(\boldsymbol{x}_L, t_1)$. However, instead of warping the predicted disparity map according to the optical flow, the gradient of the disparity is used as in (11) in order to position the predicted disparity map correctly.
4. The region based prediction $d_{P4}(\boldsymbol{x}_L, t_1)$ as described in section 2.3.

We estimate the optical flow in the central and the right image sequence. We use the method of Lucas and Kanade [9] for the estimation of a dense flow field and the wavelet based method of Bernard [3] for the estimating a single flow vector for each 16×16 block. We used the 4 abovementioned methods

Fig. 6. The difference between the next disparity frame estimated using correlation and the the predicted disparity. Left: Region based method. Right: mothod 1.

for disparity prediction on a sequence of 12 frames. Methods 2 and 3 were applied using the dense flow estimated by Lucas' and Kanade's method and the block wise flow estimated by Bernard's algorithm.

The prediction errors are summarized in table 1. Figure 6 shows the two difference images obtained

Method	M. abs. err. (pxl's)	M. rms (pxl's)
1	1.75	2.06
2 (Lucas)	1.05	1.52
2 (Bernard)	1.75	2.06
3 (Lucas)	1.05	1.53
3 (Bernard)	1.75	2.06
4	1.02	1.30

Table 1. The mean absolute error in pixels and the mean of the root of mean squares-error in pixels measured on a sequence of 12 frames.

from methods 1 and 4, respectively. The outliers in the first frame lead to high differences which cannot be predicted by any of the methods. In the other areas the difference is noticebly smaller for method 4.

3.3. Compressing the Disparity Stream

The results in Sect. 3.2 indicate that method 4, i.e., the region-based disparity propagation yields the best predictions. A second advantage of this method is the sparseness of the optical flow field. In the current experiments the algorithm generated average of 26 regions. This is crucial since the optical flow field must be transmitted together with the comressed difference image, so that a full flow field is not feasible for compression purposes. The scheme in Fig. 7 shows the update process for one disparity frame with disparity prediction from the preceding frame. In order to update one frame, we have to transmit the parameters for the regions, a list of flow-vectors (one per region), the foreground mask for the next frame, and the compressed difference between the predicted and the measured disparity. We

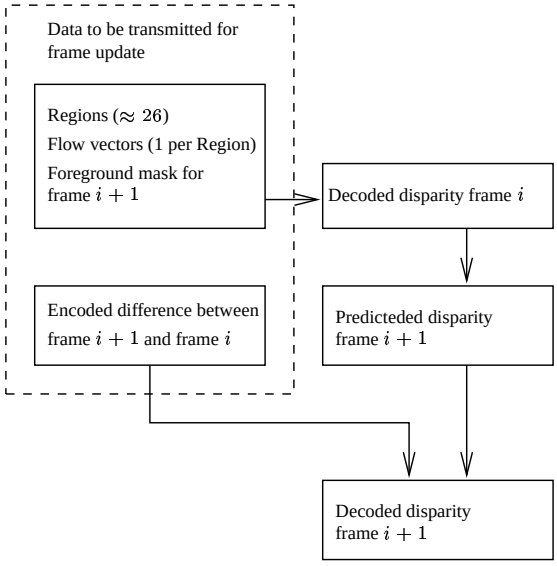

Fig. 7. Scheme of the frame update.

count the number of bits needed: A rectangular region can be identified by its center, its width and its

hight, each represented by one byte ($26 \cdot 3 \cdot 8 = 624$ bits). Each flow-vector is represented by two bytes ($26 \cdot 2 \cdot 8 = 416$ bits). The compressed of the foreground mask needed about 8000 bits in our experiments. The disparity difference images are compressed by first quantizing to 8 bpp and then compressed using JPEG 2000 with a ratio of roughly 64:1 resulting in 1300 bytes = 11 Kbits on average. This leads to an overall number of about 20 kbits to transmit per frame update. A uncompressed disparity frame is stored at an accuracy of 16 bpp at a resolution of 320×240 pixels which results in 1.23 Mbits. We thus achieve a compression ratio of 60:1 (see Fig. 8)[2].

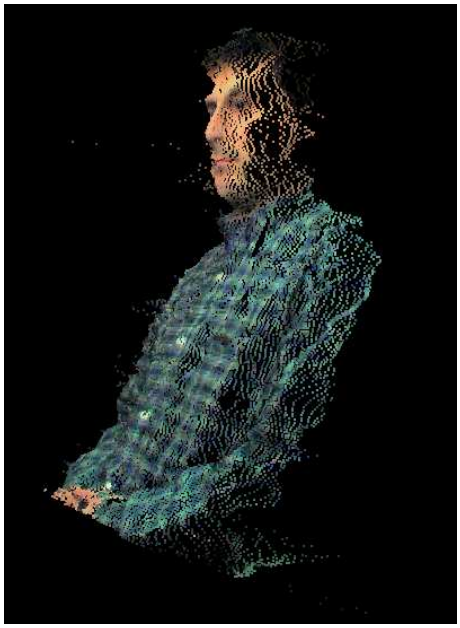

Fig. 8. A frame decoded from the previous decoded frame and the compressed difference, fbw vectors and background mask (compression ratio 60:1).

4. Conclusion

We have addressed the subject of sequences of 3D data in the context of realtime stereo reconstructions. In this scenario the data is given in form of 2D images and the 3D reconstruction is obtained from a 2D disparity map, as opposed to most computer graphics applications where complicated 3D data is given in the form of irregular meshes. The availability of highly elaborated 2D image compression schemes makes it seem reasonable to compress the 3D data based on the 2D disparity map.

We achieve a compression rate of 60:1 for the geometry data. The compression of the sequence of texture maps has not been addressed yet. It is planned to address the compression of geometry data and texture simultaneously, using the same flow fields in both cases.

[2] The movies submitted together with this paper show one uncompressed and one compressed sequence of reconstructions. The compressed sequence was generated by compressing the first frame as a single frame and all subsequent frames via region based prediction and compression of the difference images. Please note that the first four or five frames of the compressed sequence are of acceptable quality and the sequence starts to degrade for later frames due to propagation of the error.

References

1. J.L. Barron, D.J. Fleet, and S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, 1994.
2. S.S. Beauchemin and J.L. Barron. The computation of optical flow. *ACM Computing Surveys*, 27(3):433–467, 1995.
3. Ch. Bernard. Discrete wavelet analysis: a new framework for fast optic flow computation. In *European Conference on Computer Vision (ECCV'98)*, Freiburg, Germany, June 1998.
4. R.A. DeVore, B. Jawerth, and B.J. Lucier. Image compression through wavelet transform coding. *IEEE Transactions on Information Theory*, 38(2):719 –746, March 1992.
5. A. Gueziec, G. taubin, B. Horn, and F. Lazarus. A framework for streaming geometry in vrml. *CGA*, March/April 1999.
6. A. Khodakovsky, P. Schröder, and W. Sweldens. Progressive geometry compression. In *SIGGRAPH*, pages 271–278, 2000.
7. J. Lanier. Virtually there. *Scientific American*, pages 66–75, April 2001.
8. D.J. Le Gall. The MPEG video compression algorithm. *Signal Processing: Image Communication*, 4(2):129–140, 1992.
9. B.D. Lucas and T. Kanade. An iterative image-registration technique with an application to stereo vision. In *IJCAI*, pages 674–678, Vancouver, British Columbia, 1981.
10. S. Mallat. *A wavelet tour of signal processing*. Academic Press, 2nd edition, 1999.
11. M.W. Marcellin, M.J. Gormish, A. Bilgin, and M.P. Boliek. An overview of JPEG-2000. In *Data Compression Conference*, 2000.
12. J. Mulligan and K. Daniilidis. Trinocular stereo for non-parallel configurations. In *15th International Conference on Pattern Recognition*, Barcelona, Spain, September 2000.
13. A. Puri, R.V. Kollartis, and B.G. Haskell. Basics of stereoscopic video, new compression results with mpeg-2 and a proposal for mpeg-4. *Signal Processing: Image Communication*, 10:201 – 234, 1997.
14. David F. Rogers. *Procedural Elements for Computer Graphics*. WCB/McGraw-Hill, Boston, MA, second edition, 1998.
15. M. W. Siegel, S. Sethuraman, J.S. McVeigh, and Jordan A.G. Compression and interpolation of 3d-stereoscopic and multi-view video. In *Stereoscopic Displays and Virtual Reality Systems IV.*, volume 3012 of *Proc. SPIE*, pages 227 – 238, 1997.
16. S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In *International Conference on Computer Vision*, pages 722 – 729, Corfu, Greece, September 1999.