REAL TIME TRINOCULAR STEREO FOR TELE-IMMERSION

Jane Mulligan and Kostas Daniilidis

University of Pennsylvania, GRASP Laboratory
3401 Walnut Street, Philadelphia, PA 19104-6228
{janem,kostas}@grip.cis.upenn.edu

ABSTRACT

Tele-immersion is a technology that augments your space with real-time 3D projections of remote spaces thus facilitating the interaction of people from different places in virtually the same environment. Tele-immersion combines 3D scene recovery from computer vision, and rendering and interaction from computer graphics. We describe the real-time 3D scene acquisition using a new algorithm for trinocular stereo. We extend this method in time by combining motion and stereo in order to increase speed and robustness.

1. INTRODUCTION

In this paper we describe our contribution to the realization of a new mixed reality medium called tele-immersion. Tele-immersion enables users in physically remote spaces to collaborate in a shared space that mixes the local with the remote realities. The concept of tele-immersion involves all visual, aural, and haptic modalities. To date, we have dealt only with the visual part, and in collaboration with the University of North Carolina (Henry Fuchs and co-workers) and Advanced Network and Services (Jaron Lanier), we have accomplished a significant step toward realization of visual tele-immersion.

Our accomplishment is best illustrated in Fig. 1 taken during the first full scale demonstration at the University of North Carolina. A user wears passive polarized glasses and an optical tracker (Hiball) which captures the head's pose. On the two walls two realities, from Philadelphia and Armonk respectively, are stereoscopically displayed by polarized pairs of projectors. The static parts of the two scenes are view-independent 3D descriptions acquired off-line. The 3D-descriptions of the persons in the fore-ground are acquired in real-time at the remote locations and transmitted over the network. The projections on the walls are dynamically rendered according to the local user's viewpoint, and updated by real-time real-world reconstructions to increase the feeling of sharing the same conference table.

Fig. 1. A user in Chapel Hill wearing polarized glasses and an optical tracker communicates with two remote users from Philadelphia (left) and Armonk (right). The stereoscopically displayed remote 3D-scenes are composed from incoming streams of textured 3D data depicting the users, and off-line acquired static backgrounds.

There are two alternative approaches in remote immersion technologies we did not follow. The first involves video-conferencing in the large: surround projection of 2D panoramic images. This requires only a correct alignment of several views, but lacks the sense of depth and practically forbids any 3D-interaction with virtual/real objects. The second technology is closer to ours [1] but uses 3D-graphical descriptions of the remote participants (avatars). In the system description which follows, the reader will realize that such a technique could be merged with our methods in the future if we extract models based on the current raw depth points. This is just another view of the model-free vs model-based extrema in the 3D-descriptions of scenes or the bottom-up vs top-down controversy. Assuming that we have to deal with persons, highly detailed human models might be applied or extracted in the future. However, the state of avatar-based tele-collaboration is still on the level of cartoon-like representations.

In this paper, we will describe the real-time 3D acqui-

The financial support by Advanced Networks and Services and ARO/MURI-DAAH04-96-1-0007, NSF-CISE-CDS-97-03220, DARPA-ITO-MARS-DABT63-99-1-001 is gratefully acknowledged. We thank Jaron Lanier, Henry Fuchs, and Ruzena Bajcsy for their wonderful leadership in this project and Herman Towles, Wei-Chao Chen, Ruigang Yang (UNC) and Amela Sadagic (Advanced Networks and Serv.) for the so productive collaboration.

0-7803-6725-1/01/$10.00 ©2001 IEEE

sition of the dynamic parts of a scene which in Fig. 1 are the persons in the foreground. The approach we chose to follow is *view-independent* scene acquisition. Having acquired a 3D scene snapshot at a remote site, we transmit it represented with respect to a world coordinate system. Display from a new point of view involves only primitive transformations hard-wired in every graphics processor.

We will not review the huge number of existing papers (see the annual bibliographies by Azriel Rosenfeld) on all aspects of stereo. The closest approach to ours is the virtualized reality system by Narayanan and Kanade [2]. Although terms like virtualized reality and augmented reality are used in many reconstruction papers, it should be emphasized that we address a *reactive* telepresence problem, whereas most image based rendering approaches try to replace a static graphical model with a real one *off-line*. Stereo approaches may be classified with respect to the matching as well as with respect to the reconstruction scheme. Regarding matching we differentiate between sparse feature based reconstructions (see treatise in [3]) and dense depth reconstructions [4, 2, 5].

2. SYSTEM OVERVIEW AND ALGORITHMS

A tele-immersion telecubicle is designed both to acquire a 3D model of the local user and environment and to provide an immersive experience for the local user. The acquired model is used for rendering and interaction at remote sites. Immersive display is achieved via head tracking and stereoscopic projections on large scale viewscreens. The current set-up is one-way and the acquisition site consists only of a camera cluster which provides a set of trinocular views. Both responsiveness and quality of depth data are critical for immersive applications. Our system uses rectification, background subtraction, correlation matching and median filtering to balance quality and speed in our reconstructions.

2.1. Background Subtraction

Our expectation for tele-immersion is that the workspace will contain a person in the foreground interacting with remote users, and a background scene which will remain more or less constant for the duration of a session. Under this assumption we reconstruct the background scene in advance of the session and transmit it once to the remote sites. During a session, we need a method to segment out the static parts of the scene. We have chosen to implement the background subtraction method proposed by Martins et al. [6]. To further optimize calculation we compute the foreground mask for both images of the reference pair. In this way foreground pixels are only matched against foreground pixels, not background pixels.

A sequence of N (2 or more) background images B_i are acquired in advance of each session. From this set we compute a pixelwise average background image, we then compute the average pixelwise difference between the mean image and each background image $\overline{D}$ (a kind of standard deviation). During a tele-immersion session each primary image I is subtracted from the static mean background $I_D = \overline{B} - I$, a binary image is formed via the comparison $I_B = I_D > T \times \overline{D}$ where T is a configurable threshold (generally we use $T = 7$). A series of erosions and dilations are performed on I_B in order to sharpen the background mask.

2.2. Matching

The reconstruction algorithm begins by grabbing images from 2 or 3 strongly calibrated cameras. The system rectifies the images so that their epipolar lines lie along the horizontal image rows so that corresponding points lie on the same image lines, thus simplifying the search for correspondences.

In our efforts to maintain speed and quality in dense stereo depth maps we have examined a number of correlation correspondence techniques. In particular we have focussed on binocular and trinocular Sum of Absolute Differences (SAD), and binocular and trinocular Modified Normalized Cross Correlation (MNCC). In general the SAD calculation is: $corr_{SAD}(I_L, I_R) = \sum_W |I_L - I_R|$ for a window W in rectified images I_L and I_R. The disparity d determines the relative window position in the right and left images. A better correspondence metric is modified normalized cross-correlation (MNCC), $corr_{MNCC}(I_L, I_R) = \frac{2\,\mathrm{COV}(I_L, I_R)}{\sigma^2(I_L) + \sigma^2(I_R)}$.

For each pixel (u, v) in the reference image, the metrics above produce a correlation profile $c(u, v, d)$ where disparity d ranges over acceptable integer values. Selected matches are maxima (for MNCC) or minima (for SAD) in this profile. The trifocal constraint is a well known technique to refine or verify correspondences and improve the quality of stereo range data. It is based on the fact that for a hypothesized match $[u, v, d]$ in a pair of images, there is a unique location we can predict in the third camera image where we expect to find evidence of the same world point. Following Okutomi and Kanade's observation [4], we optimize over the sum of correlation values with respect to the true depth value rather than disparity. Essentially we treat the camera triple $\langle L, C, R\rangle$ as two independent stereo pairs $\langle L, C_L\rangle$ and $\langle C_R, R\rangle$, using the $\langle L, C_L\rangle$ pair to verify matches in the right-reference pair $\langle C_R, R\rangle$.

To calculate the sum of correlation scores we precompute a lookup table of the location in C_L corresponding the current pixel in C_R (based on the right-left rectification relationship). We also compute a linear approximation for the disparity $\widehat{d_L} = M(u_{C_R}, v_{C_R}) \times d_R + b(u_{C_R}, v_{C_R})$ at $[u_{C_L}, v_{C_L}]$ which arises from the same depth point as

Step	SAD	MNCC	Tri-SAD	Tri-MNCC
Capture	89	83	69	94
Rectify	25	25	48	48
Background	32	32	32	32
Matching	93	160	400	455
Median Filt.	9	9	9	9
Reconstruct	5	4	5	4
Transmit	7	7	7	7
Total	260 ms	320 ms	570 ms	650 ms
fps	3.8	3.1	1.75	1.5

Table 1. Timings for online implementations of correlation methods for 320x240 images, 60 disparities.

$[u_{C_R}, v_{C_R}, d_R]$. As we calculate the correlation score $corr_R(u_{C_R}, v_{C_R}, d_R)$, we look up the corresponding $[u_{C_L}, v_{C_L}]$ and compute $\widehat{d_L}$, then calculate the correlation score $corr_L(u_{C_L}, v_{C_L}, \widehat{d_L})$. We select the disparity d_R which optimizes $corr_T = corr_L(u_{C_L}, v_{C_L}, \widehat{d_L}) + \text{corr}_R(u_{C_R}, v_{C_R}, d_R)$. The method can be summarized as follows:

Pixelwise Trinocular Stereo

Step 1: Precompute lookup table for C_L locations corresponding to C_R locations, and approximation lookup tables M and b
Step 2: Acquire image triple $\langle L, C, R\rangle$
Step 3: Rectify $\langle L, C_L\rangle$ and $\langle C_R, R\rangle$ independently.
Step 4: Calculate foreground mask for C_R and R
Step 5: for every foreground pixel
  Step I: $corr_{best} = INVALID$, $d_{best} = INVALID$
  Step II: for every disparity $d_R \in D_r$
    Step i: compute $corr_R(u_{C_R}, v_{C_R}, d_R)$
    Step ii: lookup $[u_{C_L}, v_{C_L}]$
    Step iii: compute $\widehat{d_L} = M(u_{C_R}, v_{C_R}) \times d_R + b(u_{C_R}, v_{C_R})$
    Step iv: compute $corr_L(u_{C_L}, v_{C_L}, \widehat{d_L})$
    Step v: $corr_T = corr_L + corr_R$
    Step vi: if $corr_T$ better than $corr_{best}$ $corr_{best} = corr_T$ $d_{best} = d_R$
Step 6: Goto 2
Step 7: Median filter disparity map

3. PERFORMANCE AND RESULTS

As one would expect, methods exploiting SAD were faster than MNCC based implementations. Timings for various methods all implemented on a quad-Pentium-III-550MHz are presented in Table 1. Systems such as the Digiclops by Point Grey Research (www.ptgrey.com) and the Small Vision System (www.videredesign.com) also offer real-time stereo performance using SAD. Our timings reflect disparity ranges about twice as long as those published for these systems, but we distribute processing on a 4 processor system. We have not had the opportunity to directly compare performance and image quality,

For tele-immersion we are further interested in the quality and density of depth points. Although the computation times were greater, the high quality of trinocular depth maps makes them a desirable alternative to faster but noisier SAD range images. Figure 2 illustrates a trinocular triple and Figure 3 (a) and (b) the resulting rendered depth maps for binocular MNCC (right pair) and trinocular MNCC respectively. The improvement in depth map from use of the trinocular constraint is evident in the reduction of noise speckle and refinement in profile detail.

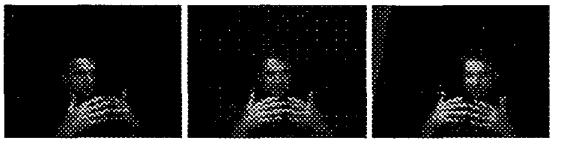
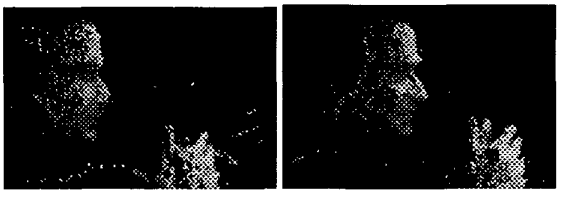


Fig. 2. Trinocular triple.



a. b.

Fig. 3. Rendered reconstructions, profile view. (a) Binocular MNCC; (b) trinocular MNCC.

4. MOTION-BASED ENHANCEMENTS

The dominant cost in stereo reconstruction is that of the correlation match itself, in general proportional to $N \times M \times D$ for images of size $N \times M$ and D tested disparity values. By using background subtraction in our application we have reduced the number of pixels considered by the search to one half to one third of the total $N \times M$. To reduce the matching costs further, we must reduce D, the number of disparities considered for each of the remaining pixels. A further observation regarding online stereo reconstruction is that for high frame rates there will be considerable similarity between successive images. We can exploit this temporal coherence in order to further optimize our online calculations. We propose a simple segmentation of the image, based on finding regions of the disparity image which contain only a narrow range of disparity values. Using a per region optical flow calculation we can estimate the location of the region in future frames, and bound its disparity search range D_i.

Our method for integrating disparity segmentation and optical flow can be summarized in the following steps, illustrated in Figure 4:

Step 1: Bootstrap by calculating full disparity map for the first stereo pair of sequence (Fig. 4(a-b)).
Step 2: Use flood-fill to segment the disparity map into rectangular windows containing a narrow range of disparities (Fig. 4(c)).

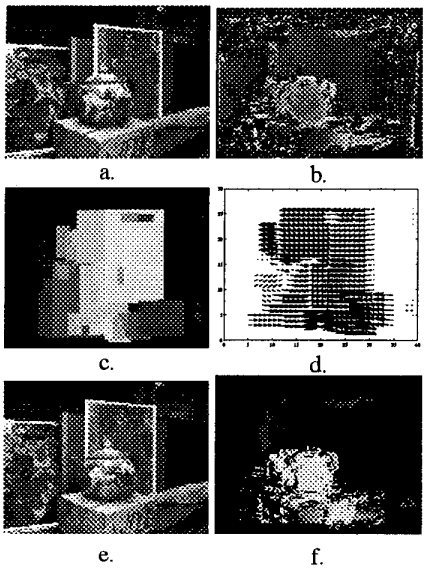


Fig. 4. Frame 25 (left) of stereo sequence (a), computed full disparity image (b), 44 extracted regions (c), flow per region (d), frame 38 (e), region based diaprity (f).

Step 3: Calculate optical flow per window for left and right smoothed, rectified image sequences of intervening frames (Fig. 4(d)).

Step 4: Adjust disparity window positions, and disparity ranges according to estimated flow.

Step 5: Search windows for correspondence using assigned disparity range, selecting 'best' correlation value over all windows and disparities associated with each pixel location (Fig. 4(e-f)).

Step 6: Goto Step 2.

Most time-critical systems using correlation matching will benefit from this approach as long as the expense of propagating the windows via optical flow calculations is less than the resulting savings over the full image/full disparity match calculation.

Restricting the change in disparity per window essentially divides the underlying surfaces into patches where depth is nearly constant. We use a threshold on the maximum absolute difference in disparity as the constraint defining regions, and we allow regions to overlap. Only rectangular image windows are maintained, rather than a convex hull or more complicated structure, because it is generally faster to apply operations to a larger rectangular window than to manage a more complicated region structure. Regions are extracted using flood fill or seed fill, a simple polygon filling algorithm from computer graphics.

Optical flow calculations approximate the motion field of objects moving relative to the cameras, based on the familiar image brightness constancy equation: $I_x v_x + I_y v_y + I_t = 0$, where I is the image brightness and I_x, I_y and I_t are the partial derivatives of I with respect to x, y and t, and $v = [v_x, v_y]$ is the image velocity. We use a standard local weighted least square algorithm [7] to calculate values for v based on minimizing $e = \sum_{W_i}(I_x v_x + I_y v_y + I_t)^2$, for the pixels in the current window W_i. For each disparity window we assume the motion field is constant across the region W_i, and calculate a single value for the centre pixel. Given image regions, we must now adjust their location according to our estimated flow for the right and left images. Basically we force the window to expand rather than actually moving it. Since the windows have moved as a consequence of objects moving in depth, we must also adjust the disparity range $D(t) = [d_{min}, d_{max}]$ for each window using the estimated flow velocities.

In the case of our disparity windows, each window can be of arbitrary size, but will have relatively few disparities to check. Because our images are rectified to align the epipolar lines with the scanlines, the windows will have the same y coordinates in the right and left images. Given the disparity range we can extract the desired window from the right image given $x_r = x_l - d$.

The complexity of stereo correspondence on our proposed window system is about half that for full images, depending on the number of frames in time used to estimate optical flow. We have demonstrated experimentally that our window-based reconstructions compare favourably to those generated by correlation over the full image, even after several frames of propagation via estimated optical flow. The observed mean differences in computed disparities were less than 1 pixel and the maximum standard deviation was 4.4 pixels.

5. REFERENCES

[1] J. Leigh, A.E. Johnson, M. Brown, D.J. Sandin, and T.A. DeFanti, "Visualization in teleimmersive environments," *Computer*, vol. 32, no. 12, pp. 66–73, 1999.

[2] P. Narayanan, P. Rander, and T. Kanade, "Constructing virtual worlds using dense stereo," in *Proc. ICCV*, 1998, pp. 3–10.

[3] O. Faugeras, *Three-dimensional Computer Vision*, MIT-Press, Cambridge, MA, 1993.

[4] Masatoshi Okutomi and Takeo Kanade, "A multiple-baseline stereo," *IEEE PAMI*, vol. 15, no. 4, pp. 353–363, April 1993.

[5] F. Devernay and O. Faugeras, "Computing differential properties of 3-d shapes from stereoscopic images without 3-d models," in *Proc. CVPR*, Seattle, WA, June 1994, pp. 208–213.

[6] Fernando C. M. Martins, Brian R. Nickerson, Vareck Bostrom, and Rajeeb Hazra, "Implementation of a real-time foreground/background segmentation system on the intel architecture," in *IEEE ICCV99 Frame Rate Workshop*, Kerkyra, Greece, Sept. 1999.

[7] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *DARPA Image Understandig Workshop*, 1981, pp. 121–130.