University of Pennsylvania
ScholarlyCommons

Technical Reports (CIS) | Department of Computer & Information Science

October 2000

View-independent Scene Acquisition for Tele-Presence

Jane Mulligan
University of Pennsylvania

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

Follow this and additional works at: http://repository.upenn.edu/cis_reports

Recommended Citation
Mulligan, Jane and Daniilidis, Kostas, "View-independent Scene Acquisition for Tele-Presence" (2000). *Technical Reports (CIS)*. Paper 70.
http://repository.upenn.edu/cis_reports/70

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-00-16.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/cis_reports/70
For more information, please contact repository@pobox.upenn.edu.

View-independent Scene Acquisition for Tele-Presence

Abstract
Tele-immersion is a new medium that enables a user to share a virtual space with remote participants. The user is immersed in a rendered 3D-world that is transmitted from a remote site. To acquire this 3D description we apply bi- and trinocular stereo techniques. The challenge is to compute dense stereo range data at high frame rates, since participants cannot easily communicate if the processing cycle or network latencies are long. Moreover, new views of the received 3D-world must be as accurate as possible. We address both issues of speed and accuracy and we propose a method for combining motion and stereo in order to increase speed and robustness.

Comments
University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-00-16.

This technical report is available at ScholarlyCommons: http://repository.upenn.edu/cis_reports/70

View-independent Scene Acquisition for Tele-Presence

Jane Mulligan and Kostas Daniilidis
University of Pennsylvania, GRASP Laboratory*
3401 Walnut Street, Philadelphia, PA 19104-6228
{janem,kostas}@grip.cis.upenn.edu
COMPUTER AND INFORMATION SCIENCE DEPARTMENT
TECHNICAL REPORT MS-CIS-00-16
Appears also in the
INTERNATIONAL SYMPOSIUM OF AUGMENTED REALITY, MUNICH, OCT. 5-6, 2000

Abstract

Tele-immersion is a new medium that enables a user to share a virtual space with remote participants. The user is immersed in a rendered 3D-world that is transmitted from a remote site. To acquire this 3D description we apply bi- and trinocular stereo techniques. The challenge is to compute dense stereo range data at high frame rates, since participants cannot easily communicate if the processing cycle or network latencies are long. Moreover, new views of the received 3D-world must be as accurate as possible. We address both issues of speed and accuracy and we propose a method for combining motion and stereo in order to increase speed and robustness.

1 Introduction

The power of today's general purpose and graphics processors and the high bandwidth of the recent Internet generations, provide the necessary infrastructure for mixed reality systems which can augment the user's senses and create the sense of tele-presence. 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 senses. 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.

Figure 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.

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 [34, Hiball] which captures the head's pose. On the two walls two realities, the Philadelphia, and the Armonk real-

*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 Netw. and Serv.) for the so productive collaboration.

ity, respectively, are stereoscopically displayed from 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. This world-wide first presentation speaks to the feasibility of what Raymond Kurzweil [9] predicts for the year 2019:

> *You can do virtually anything with anyone regardless of physical proximity... Phone calls routinely include high-resolution three-dimensional images projected through the direct-eye displays and auditory lenses. Three-dimensional holography displays have also emerged. In either case, users feel as if they are physically near the other person... Thus a person can be fooled as to whether or not another person is physically present or is being projected through electronic communication. The majority of "meetings" do not require physical proximity.*

There is still a long way to go to achieve a compelling "fooling" of the user's senses. However, people can already start testing this new medium for tele-collaboration. Participants in the experiments felt that the tele-immersive environment was superior to conventional video-conferencing.

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 [12] and 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.

Comparing tele-immersion to classical augmented reality we find that real-time head tracking and display refresh rate pose minor problems. The challenging difference is, first, that the display used is a spatially augmented display [26] and not an HMD and, second, that the mixed components are not pre-stored perfect virtual objects, but on-line acquired real range data. In addition these data are transmitted over the network before displayed. Sense of presence really depends on real-time responses and accurate depth estimation with respect to the viewer. In this paper, we will describe the real-time 3D acquisition 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 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. In addition to real time, we want the new view to be error free so that the user does not experience wrong depths through her polarized stereo glasses. The basic question is how to achieve a perceptually best reconstruction in real-time. We have to emphasize that these criteria are stricter than those in navigation, for example. Navigational stereo targets a convex-hull based representation whereas the user here must be able to see features as detailed as a face profile reconstructed from frontal views.

2 Related Acquisition Work

We will not review the huge number of existing papers (see the annual bibliographies by Azriel Rosenfeld) on all aspects of stereo (the reader is referred to a standard review [5]). Application of stereo to image based rendering is very well discussed and reviewed in the recent paper by Narayanan and Kanade [23]. 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 [6]) and dense depth reconstructions [25, 23]. Approaches such as [4, 31] address the probabilistic nature of matching with particular emphasis on the occlusion problem. Area-based approaches [15] are based on correlation and emphasize the real-time re-

sponsiveness as we do. An approach with emphasis on virtualized reality is [23]. This system captures the action of a person from a dome of 51 cameras. Surround camera clusters are also very suitable for voxel-based techniques like space-carving [10, 29, 18]. The processing is off-line and in this sense there is no indication of how it could be used in telepresence beyond the off-line reconstruction of static structures.

With respect to reconstruction, recent approaches can be classified as strongly or weakly (or self-calibrated) approaches. Self-calibration approaches [17] provide a metric reconstruction from multiple views with an accuracy which is suitable only for restricted augmented reality applications like video manipulation where the quality of depth is not relevant. Weakly calibrated approaches [11] provide real time performance and are suitable for augmenting scenes only with synthetic objects.

Recently, gaining insight from the afore-mentioned projective geometric work, the paradigm of view generation became popular. Instead of calibrating the cameras with respect to a world coordinate system the fundamental matrices of image pairs are computed. Then the images are rectified and dense disparity maps are build. A new view can be computed by constructing the fundamental matrices with respect to each given view [8, 28]. We believe that besides calibration there is no real gain in using fundamental instead of projection matrices. The real bottleneck of establishing a depth disparity map exists in both. To alleviate the correspondence problem view-dependent approaches were established which are calibrated but rely on the silhouettes of objects [16].

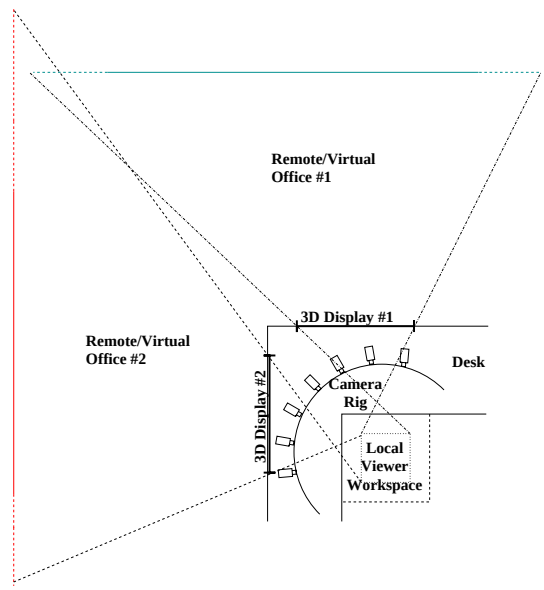


Figure 2. Tele-cubicle Camera configuration

Figure 3. Camera configuration, user view.

3 System's Overview and Architecture

A tele-immersion telecubicle is designed both to acquire a 3D model of the local user and environment for rendering and interaction at remote sites, and to provide an immersive experience for the local user via head tracking and stereoscopic display projected on large scale viewscreens. A typical setup is depicted in Figure 2. The user moves freely in a 1 m workspace at his desk. Remote users are rendered on 90cm$\times$120cm screens by projector pairs. The user wears lightweight polarized glasses and a head-tracker to drive the stereo display function.

A cluster of 7 firewire cameras are arranged on an arc at 15° separation to 'surround' the user and prevent any break presence due to a hard edge where the reconstruction stops. In the current set-up none of the participating sites has a full version of the telecubicle. Instead, the display site is as illustrated and described in Figure 1 and the acquisition site consists only of a camera cluster as illustrated in Figure 3. These cameras are used to calculate binocular or trinocular stereo depth maps from overlapping pairs or triples. For example the combined trinocular reconstruction illustrated in Figure 10, was computed from 5 triples $\langle C_0, C_1, C_2\rangle$, $\langle C_1, C_2, C_3\rangle$, $\langle C_2, C_3, C_4\rangle$, $\langle C_3, C_4, C_5\rangle$, and $\langle C_4, C_5, C_6\rangle$. The technical obstacle to the combining of camera views, is that each reconstruction is performed on a separate computer. Since the digital firewire cable is not 'splitable' we are forced to transmit images to neighbouring machines via Ethernet/TCP/IP, adding about 130 ms to the overhead of the system.

Both responsiveness and quality of depth data are critical for immersive applications. In order improve the frame rate of our system we have applied a number of techniques to reduce the weight of calculation, particularly in the expensive correlation matching re-

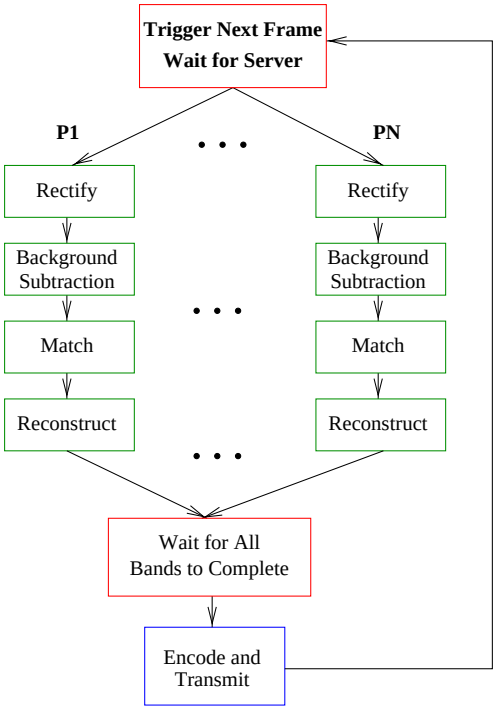


Figure 4. Parallelized system.

quired to generate dense depth maps. The simplest technique for the developer of course, is to purchase more and faster computers. We have built our system on 5 quad PIII 550 MHz servers (one for each reconstructed view) and parallelized our code accordingly.

The general parallel structure of the system is illustrated in Figure 4. One of the servers acts as a trigger server for the firewire acquisition. When all of the reconstructors are ready for the next frame the trigger server triggers all of the cameras simultaneously. Each computer grabs the image from 1 or 2 cameras and transmits and receives the images needed by its neighbours and itself. Within each quad machine the images are divided into 4 equal bands and each processor is devoted to a particular band. The thread for each processor rectifies, background subtracts, matches and reconstructs points in its band of the image. When all processors have completed processing the texture and depth map are transmitted via TCP/IP to a remote renderer. This data is encoded as 3-(320×240) unsigned char image planes (RGB) of texture, plus one unsigned short image plane where $1/z$ values have been scaled into unsigned short, and background and unmatched foreground pixels are flagged. The total is about 3 Mbits per view per frame.

3.1. 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. To obtain the speed and quality of depth points our application requires, we reconstruct the background scene in advance of the session and transmit it once to the remote sites. While the user moves in the foreground 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. [14].

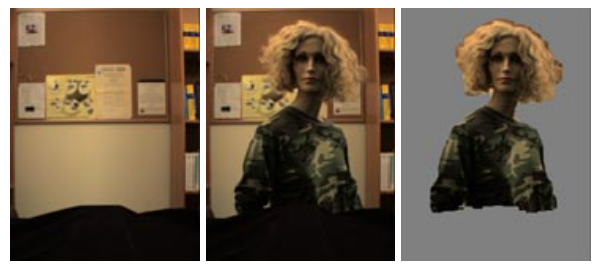


Figure 5. Background image, foreground image and subtracted result.

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 $\overline{B} = \frac{1}{N}\sum_i B_i$. We then compute the average pixelwise difference between $\overline{B}$ and B_i, $\overline{D} = \frac{1}{N}\sum_i(\overline{B} - B_i)$.

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 $T = 7$). These thresholded difference images are quite noisy. A series of erosions and dilations are performed on I_B in order to sharpen the background mask. The morphological operations are implemented by separable convolutions with masks of ones of size W. For erosion, if a pixel has a value less than $W \times W$ after convolution, it is set to zero; for dilation, if a pixel has a value greater than zero it is set to one. The sequence is as follows: erode $W = 3$, dilate $W = 7$, erode $W = 17$, dilate $W = 9$. Typical results are illustrated in Figure 5.

3.2. Bi- and Trinocular Matching

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 Sum of Absolute Differences (SAD), because of the speed provided by hardware specific operations, and Modified Normalized Cross Correlation (MNCC), which we have found produces superior depth maps in the binocular case. We have also tested trinocular SAD and MNCC for our system, which is slightly more challenging in a surround camera configuration such as ours.

3.2.1 Correlation Methods

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 to reduce the search space for correspondences, and so that corresponding points lie on the same image lines.

The calculation of SAD as a correlation metric is facilitated on Intel/MMX machines by the assembler operation **psadbw** which can calculate the sum of absolute differences between two registers containing 8 char values each, in a single cycle. In general the SAD calculation is:

$$corr_{SAD}(I_L, I_R) = \sum_W |I_L - I_R| \qquad (1)$$

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),

$$corr_{MNCC}(I_L, I_R) = \frac{2\ \mathrm{cov}(I_L, I_R)}{\sigma^2(I_L)\ +\ \sigma^2(I_R)}. \qquad (2)$$

where I_L and I_R are the left and right rectified images over the selected correlation windows.

For each pixel (u, v) in the left 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.

3.2.2 Non-parallel Trinocular Configurations

The trinocular epipolar 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 [5]. A hypothesis is correct if the epipolar lines for the original point $[u, v]$

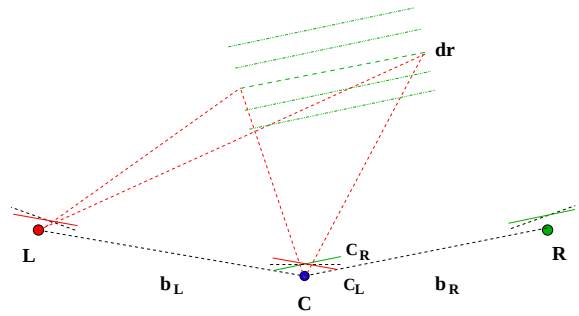


Figure 6. Trinocular camera triple.

and the hypothesized match $[u - d, v]$, intersect in the third camera image. The most common scheme for exploiting this constraint is to arrange the camera triple in a right angle, allowing matching along the rows and columns of the reference image [24, 1, 7, 21]. Our telecubicle configuration, illustrated in Figure 2, does not allow us to arrange or rectify triples of camera image planes such that they are coplanar, and therefore it is more expensive for us to exploit the trinocular constraint.

Following Okutomi and Kanade's observation [25], 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 $\langle C_R, R\rangle$. Figure 6 illustrates the general premise: any disparity for the reference pair $\langle C_R, R\rangle$, represents a surface of constant depth (with respect to that pair) in the world, however for the left pair $\langle L, C_L\rangle$ this surface involves a range of distances and therefore disparities.

In previous work [20] we explored two approaches to exploiting the trinocular constraint in surround camera configurations. The first method we used precomputed correlation images for ranges of disparity in the left camera pair, then the computed correlation for each tested $[u_R, v_R, d_R]$ was added to that precomputed for the corresponding $[u_L, v_L, d_L]$. This results in large correlation lookup tables for the left image pair.

The second method was an attempt to avoid large lookup tables by independently finding the best N extrema in the correlation surfaces for both image pairs. These sorted hypotheses were then cross checked to determine whether a common depth point gave rise to the scores for any pair. Valid hypothesis pairs with the best score were retained. This method required less lookup table space, but had considerable added overhead to maintain the sorted hypotheses.

When revising our system design to parallelize and improve its speed, we discovered that by using fore-

ground segmentation we need consider only one half to one third of the pixels in the reference image C_R. This makes it feasible to calculate the entire correlation profile for each pixel one at a time. 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 $[u_{C_R}, v_{C_R}, d_R]$. 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

$$corr_T = corr_L(u_{C_L}, v_{C_L}, \widehat{d_L}) + \mathrm{corr}_R(u_{C_R}, v_{C_R}, d_R)$$

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

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

4. Performance and Results

As one would expect methods exploiting SAD were faster than MNCC based implementations. All implementations ran on a quad-PIII 550 MHz server in 1 second or less, including image acquisition and transfer and transmission of reconstructions to the renderer. Timings for the various systems are presented in Table 1.

Step	SAD	MNCC	Tri-SAD	Tri-MNCC
Rectify	49	50	49	48
Background	18	18	18	18
Matching	182	261	390	791
Reconstruct	6	6	7	6
Total	446 ms	520 ms	662 ms	1067 ms
fps	2.2	1.9	1.5	0.9

Table 1. Timings for online implementations of correlation methods. Frames per second (fps) values include 160 ms capture time and 6 ms network transmission overhead.

Figure 7. Trinocular triple.

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 7 illustrates a trinocular triple and Figure 8 (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.

An added challenge with our seven camera cluster is the combination of multiple reconstructions into a single rendered view. We currently depend on the accuracy of our calibration to a common reference frame for all cameras. Figure 9 shows a full set of camera views for a single frame in the current telecubicle camera cluster. From this image set 5 reconstructed

a. b.

Figure 8. Rendered reconstructions, profile view. (a) Binocular MNCC; (b) trinocular MNCC.

Figure 9. Seven camera views.

Figure 10. Five trinocular reconstructions combined and rendered, rotated view.

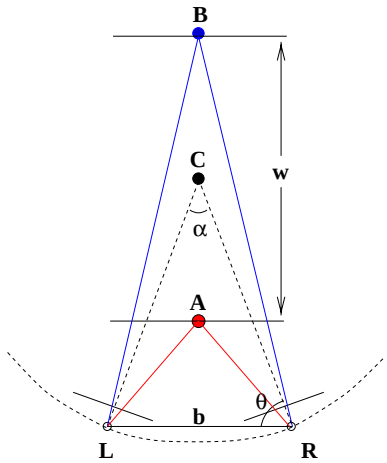


Figure 11. Verged Stereo pair configuration for work volume w.

views are calculated for overlapping triples. Figure 10 shows a profile rotation of the total set of 164,000 depth points calculated using trinocular MNCC for the frame in Figure 9.

5. 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 would like to reduce D, the number of disparities considered for each of the remaining pixels.

Consider the diagram of a verged camera pair in Figure 11. To maintain a seamless immersive experience, we cannot greatly restrict the motion of subjects in the stereo workspace. For a workspace depth $\mathbf{w} = 1\ m$, the disparity ranges from $d = -61$ pixels at point A, 75 cm from the cameras to $d = 87$ pixels at B, 175 cm from the cameras. Clearly a disparity range of $87 - (-61) = 148$ is prohibitive for an exhaustive correspondence search in real-time applications.

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.

The complimentary nature of optical flow and stereo calculations is well known. Stereo correspondence suffers from the combinatorics of searching across a range of disparities, and from occlusion and surface discontinuities. Structure from motion calculations are generally second order and sensitive to noise, as well as being unable to resolve a scale factor. Exploiting the temporal coherence of depth and flow measurements can take two forms: it can be used to improve the quality or accuracy of computed values of depth and 3D motion [22, 33, 2] or, as in our case, can be used as means of optimizing computations to achieve real-time performance. Obviously ap-

proaches which compute accurate 3D models, using iterative approaches such as linear programming are unlikely to be useful for real-time applications such as ours. Other proposed methods for autonomous robots, restrict or otherwise depend on relative motion [3, 30] which cannot be controlled for a freely moving human subject.

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

Step 1: Bootstrap by calculating a full disparity map for the first stereo pair of the sequence.

Step 2: Use flood-fill to segment the disparity map into rectangular windows containing a narrow range of disparities.

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

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

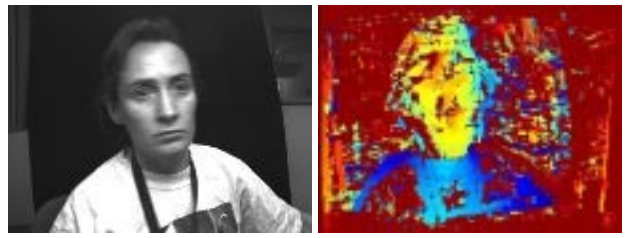
Step 5: Search windows for correspondence using assigned disparity range, selecting 'best' correlation value over all windows and disparities associated with each pixel location.

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.

Flood-fill Segmentation. Restricting the change in disparity per window essentially divides the underlying surfaces into patches where depth is nearly constant. The image of a curved surface for example will be broken into a number of adjacent windows, as will a flat surface angled steeply away from the cameras. Essentially these windows are small quasi-frontal planar patches on the surface.

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 [27, pp. 137-141], a simple polygon filling algorithm from computer graphics. We have implemented a scan-line version which pops a seed pixel



a. b.

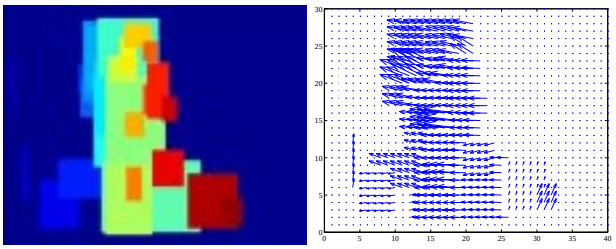
Figure 12. Frame 12 (left) of stereo sequence (a), and computed disparity image (b).

location inside a polygon to be filled, then finds the right and left connected boundary pixels on the current scan line, 'filling' those pixels between. Pixels in the same x-range in the lines above and below are then examined. The rightmost pixel in any unfilled, non-boundary span on these lines in this range, is pushed on the seed stack and the loop is repeated. When the stack is empty the polygon is filled.

We have modified this process slightly so that the boundary is defined by whether the current pixel / disparity value falls within a threshold ($+/-5$) of the first seeded pixel. We start with a mask of valid disparity locations in the disparity image. For our purposes filling is marking locations in the mask which have been included in some disparity region, and updating the upper left and lower right pixel coordinates of the current window bounding box. When there are no more pixels adjacent to the current region which fall within the disparity range of the original seed, the next unfilled pixel from the mask is used to seed a new window. Once all of the pixel locations in the mask are set the segmentation is complete.

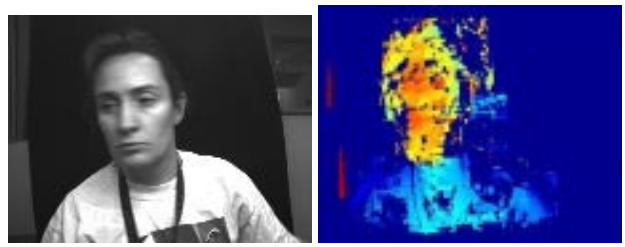
As a final step small regions are attributed to noise and deleted. Nearby or overlapping windows are merged when the difference between the region mean disparities is small. Figures 12 and 13 illustrate an image, its disparity map and the rectangular regions extracted via flood-fill.

Flow per Window 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 [13, 32] to calculate values for v



a. b.

Figure 13. Extracted windows of similar disparity (a) and calculated flow per window (b).



a. b.

Figure 14. Frame 18 of sequence (a) and region based disparity map (b).

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. Figure 13 b. illustrates the optical flow values calculated on a per window basis.

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, if the left coordinate of the window is predicted to move up or left by the right or left flow, then the window is enlarged to the left. If the right coordinate is predicted to move down or right the window is enlarged accordingly.

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:

$$D(t+dt) = [\min(d_{min} + v_{xl}dt - v_{xr}dt, d_{min}), \\ \max(d_{max} + v_{xl}dt - v_{xr}dt, d_{max})].$$

Windowed Correspondence. Window based correspondence proceeds in much the same way as described in Section 3.2.1. 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 $x_r = x_l - d$. Correlation matching and assigning valid matches to the disparity map proceeds as usual.

Figure 14 illustrates the result of propagating the disparity windows from frame 12 to frame 18 of a stereo image sequence. The subject is translating and rotating from right to left in the image. In [19] we have examined the quality and complexity tradeoffs of this approach in detail. 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.

6 Conclusion

We presented the scene acquisition part of a first testbed for visual tele-immersion. We decided to pursue a view-independent approach and constructed a 3D description with respect to a world-coordinate system. We achieved a surround view with a throughput of almost 2fps. We are looking forward into achieving 10fps without sacrificing depth quality. To eliminate outliers we will address both problems of highlights and occlusions in the near future. Our ultimate goal is to perform systematic experiments and study the sense of presence and the nuances of communication during collaboration tasks.

References

[1] N. Ayache. *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*. The MIT Press, Cambridge, MA, 1991.

[2] P. Balasubramanyam. The p-field: A computational model for binocular motion processing. In *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 115–120, 1991.

[3] P. Beardsley, J. Brady, and D. Murray. Prediction of stereo disparity using optical flow. In *British Machine Vision Conference*, pages 259–264, 1990.

[4] P. Belhumeur. A bayesian approach to binocular stereopsis. *Intl. J. of Computer Vision*, 19(3):237–260, 1996.

[5] U. Dhond and J. Aggrawal. Structure from stereo: a review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510, 1989.

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

[7] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge, MA, 1993.

[8] O. Faugeras and L. Robert. What can two images tell us about a third one? In *Proc. Third European Conference on Computer Vision*, pages 485–492. Stockholm, Sweden, May 2-6, J.O. Eklundh (Ed.), Springer LNCS 800, 1994.

[9] R. Kurzweil. *The Age of Spiritual Machines*. Penguin Press, 2000.

[10] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *Proc. Int. Conf. on Computer Vision*, pages 307–314, 1999.

[11] K. Kutulakos and J. Vallino. Calibration-free augmented reality. *IEEE Trans. on Visualization and Computer Graphics*, 4(1):1–20, 1998.

[12] J. Leigh, A. Johnson, M. Brown, D. Sandin, and T. DeFanti. Visualization in teleimmersive environments. *Computer*, 32(12):66–73, 1999.

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

[14] F. C. M. Martins, B. R. Nickerson, V. Bostrom, and R. Hazra. Implementation of a real-time foreground/background segmentation system on the intel architecture. In *IEEE ICCV99 Frame Rate Workshop*, Kerkyra, Greece, Sept. 1999.

[15] L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *International Journal of Computer Vision*, 8:71–91, 1992.

[16] W. Matusik, C. Buheler, R. Raskar, S. Gortler, and L. McMillan. Image-based visual hulls. In *ACM SIGGRAPH*, 2000. to appear.

[17] S. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8:123–151, 1992.

[18] S. Moezzi, L.-C. tai, and P. Gerard. Virtual view generation from 3d digital video. *IEEE Multimedia*, 4:18–26, 1997.

[19] J. Mulligan and K. Daniilidis. Predicting disparity windows for real-time stereo. In *Proceedings of the Sixth European Conference on Computer Vision*, Dublin, Ireland, June 2000. In Press.

[20] J. Mulligan and K. Daniilidis. Trinocular stereo for non-parallel configurations. In *Proceedings of the 15th International Conference on Pattern Recognition*, Barcelona, Spain, Sept. 2000. In press.

[21] D. Murray and J. Little. Using real-time stereo vision for mobile robot navigation. *Autonomous Robots*, 2000. to Appear.

[22] H.-H. Nagel. Dynamic stereo vision in a robot feedback loop based on the evaluation of multiple interconnected displacement vector fields. In *Robotics Research: The Third International Symposium*, pages 49–55, Gouvieux, France, October 1985.

[23] P. Narayanan, P. Rander, and T. Kanade. Constructing virtual worlds using dense stereo. In *Proc. Int. Conf. on Computer Vision*, pages 3–10, 1998.

[24] Y. Ohta, M. Watanabe, and K. Ikeda. Improving depth map by right-angled trinocular stereo. In *Proceedings of the 8th International Conference on Pattern Recognition (ICPR'86)*, volume I, pages 519–521, Paris, France, Oct. 1986.

[25] M. Okutomi and T. Kanade. A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 15(4):353–363, April 1993.

[26] R. Raskar, G.Welch, M.Cutts, A.Lake, L.Stesin, and H.Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *ACM SIGGRAPH*, pages 179–188, 1998.

[27] D. F. Rogers. *Procedural Elements for Computer Graphics*. WCB/McGraw-Hill, Boston, MA, second edition, 1998.

[28] D. Scharstein and R. Szeliski. Stereo matching with non-linear diffusion. *Proc. Int. Conf. Computer Vision and Pattern Recognition*, 1996.

[29] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1067–1073, Puerto Rico, June 17-19, 1997.

[30] M. Tistarelli, E. Grosso, and G. Sandini. Dynamic stereo in visual navigation. In *Proceedings 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 186–193, 1991.

[31] C. Tomasi and R. Manduchi. Stereo without search. *Proc. European Conf. Computer Vision*, 1996.

[32] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall Inc., Upper Saddle River, NJ, 1998.

[33] A. M. Waxman and J. H. Duncan. Binocular image flows: Steps toward stereo-motion fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):715–729, November 1986.

[34] G. Welch and G. Bishop. Scaat: Incremental tracking with incomplete information. In *ACM SIGGRAPH*, 1997.