Stereo-Based Environment Scanning for Immersive Telepresence

Jane Mulligan, Xenophon Zabulis, Nikhil Kelshikar, and Kostas Daniilidis, *Member, IEEE*

***Abstract*—The processing power and network bandwidth required for true immersive telepresence applications are only now beginning to be available. We draw from our experience developing stereo based tele-immersion prototypes to present the main issues arising when building these systems. 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 three-dimensional (3-D) world that is transmitted from a remote site. To acquire this 3-D description, we apply binocular and trinocular stereo techniques which provide a view-independent scene description. Slow processing cycles or long network latencies interfere with the users' ability to communicate, so the dense stereo range data must be computed and transmitted at high frame rates. Moreover, reconstructed 3-D views of the remote scene must be as accurate as possible to achieve a sense of presence. We address both issues of speed and accuracy using a variety of techniques including the power of supercomputing clusters and a method for combining motion and stereo in order to increase speed and robustness. We present the latest prototype acquiring a room-size environment in real time using a supercomputing cluster, and we discuss its strengths and current weaknesses.**

***Index Terms*—Stereo vision, tele-immersion, telepresence, terascale computing.**

I. Introduction

HIGH-SPEED desktop computers, digital cameras, and Internet2 connections are making collaboration via immersive telepresence a real possibility. The missing link is currently the techniques to extract, transmit, and render the information from sensors at a remote sight such that the local user has the compelling sense of "being there." Over the past six years, researchers in the GRASP Laboratory at the University of Pennsylvania have worked with the National Tele-immersion Initiative [1] to provide real-time three-dimensional (3-D) stereo reconstructions of remote collaborators for immersive telepresence systems. In this paper, we report our progress on the most important aspect of tele-immersion: the ability to acquire dynamic 3-D scenes in real time.

Any form of telepresence needs to convey information about a remote place to the local user, in an immediate and compelling manner. Both real-time update and visual quality are necessary conditions for effective remote collaboration. A remote scene has to be acquired in real time with as much real-world detail as possible. The need for using real images has been widely recognized in computer graphics, and image-based rendering is now an established area in vision and graphics.

For tele-immersion, we decided to follow a view-independent scene acquisition approach in order to decouple the rendering rate from the acquisition rate and network delays. View independence allows us to transmit the same 3-D model to many participants in a virtual meeting place, providing each immersive display with a model to rerender as the viewer moves within her augmented display. We have chosen to use dense normalized correlation stereo to capture 3-D data in order to provide dense and accurate view-independent models for rendering in the immersive display. A true, detailed 3-D model is also important for interaction with objects in the 3-D space.

In our extensive experience with tele-immersion prototypes, we have explored many aspects of capturing desktop and room-size environments using dense correlation stereo. Issues from calibrating many cameras in a large volume to how to place cameras for best accuracy in reconstruction arise in constructing working systems. We have also explored many of the possible speed–quality tradeoffs for stereo-based environment scanning including quantitative comparison to ground truth data for many aspects of our systems [2], [3].

Obviously, in this age of ever increasing CPU speed, one way to improve the speed of depth acquisition is to apply more cycles. Our latest prototype does exactly this, utilizing the power of the Pittsburg Supercomputing Center (PSC) to achieve room-size reconstructions from many 640×480 images at high frame rates. Another avenue we are exploring, in order to improve the temporal performance of reconstruction on an image sequence, is to take advantage of temporal coherence. Since the same objects tend to be visible from frame to frame—background walls and furniture stay static—we can use knowledge from earlier frames when processing new ones. However, as usual, there is a complicated tradeoff between the calculation we add for disparity segmentation and motion estimation, and the advantages of predicting disparity ranges for simplifying the stereo correspondence problem.

This paper summarizes the evolution of an immersive telepresence system and presents the highlights of today's version. The main contribution of our approach to the state of the art is in the combination of following points.

Manuscript received January 20, 2003; revised April 9, 2003. This work was supported by the National Science Foundation under Grant NSF-IIS-0083209, Grant NSF-EIA-0218691, Grant NSF-IIS-0121293, and Grant NSF-EIA-9703220, by DARPA/ITO/NGI through a subcontract from the University of North Carolina, and by a Penn Research Foundation grant.

J. Mulligan is with the Department of Computer Science, University of Colorado at Boulder, Boulder, CO 8030 USA (e-mail: Jane.Mulligan@colorado.edu).

N. Kelshikar, X. Zabulis, and K. Daniilidis are with the Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389 USA (e-mail: zabulis@grasp.cis.upenn.edu; nikhil@grasp.cis.upenn.edu; kostas@grasp.cis.upenn.edu).

Digital Object Identifier 10.1109/TCSVT.2004.823390

- It is the first integrated 3-D remote telepresence system comprising geographically distributed image acquisition, 3-D computation, and display.
- It is the first stereo system that works in a wide area without any background subtraction, implemented in real time using a supercomputer.
- We propose a temporal prediction process which decreases disparity search range on segmented image regions for faster stereo correspondence.

The remainder of the paper is organized as follows: We continue this section with an overview of telepresence, underlying challenges in acquisition, and related systems. In Section II, we review the evolution of our working tele-immersion prototypes. In Section III, we present the details of our stereo algorithm. In Section IV, we describe an experimental evaluation focusing on the effects of kernel size and view misregistration. In Section V, we propose prediction and motion-depth modeling to optimize the disparity range search.

A. Telepresence Systems

Telepresence systems can generally be viewed as composed of three parts: a capture system to record and represent the information from the remote site, a network transmission system, and a display system to make the local user feel as if she were somehow present in the remote scene. These three parts become even more challenging if we require telepresence to be immersive, which means to create the illusion of being in an environment different than the viewer's true physical surroundings.

The first question is, how do we capture representations of the remote scene that are adequate for the task of creating a believable remote presence? We have chosen *view-independent* 3-D acquisition with correlation-based stereo because it is fast and noninvasive. The representation has to be view-independent so that rendering can be decoupled and thus asynchronous to acquisition. A second advantage of view independence is that an acquired 3-D representation can be broadcast to several receivers. In contrast, a view-dependent approach requires either sending all images to the remote site where the novel view is computed or computing views locally and transmitting them. In the latter case, receiving feedback about the user's head position through the network would cause an unmitigated latency.

There are two alternative approaches in remote immersion technologies we did not follow. The first involves video conferencing in the large: surround projection of two-dimensional (2-D) panoramic images. This requires only a correct alignment of several views, but lacks the sense of depth and practically forbids any 3-D interaction with virtual/real objects. The second technology [4] uses 3-D graphical descriptions of the remote participants (avatars). This is just another view of the model-free versus model-based extrema in the 3-D description of scenes or the bottom-up versus top-down controversy. Assuming that we have to deal with persons, human models might be applied [5] combined with image-based rendering, but everything in a scene has to be scanned prior to a telepresence session and fine detail is still missing. Of course, errors in model-based approaches take the form of outlier poses of the avatar, as opposed to outlier depth points or holes in stereo.

The networking aspects of telepresence systems are just beginning to be defined. If there is communication and interaction involved, then latency is the most critical issue. Bandwidth affects the frame arrival rate. If a lossy protocol is applied, we need techniques to recover from losses, both on the way from the cameras to the computing resources as well as on the way from the computing resources to the display. If compression is applied, then again we need a special image compression on the way from the camera to the computer which minimizes decimation in stereo matching, and another 3-D compression on the way to the display.

Finally, regardless of the speed and quality of the data arriving at the display side, if the local user's viewing environment is not updated smoothly and quickly enough by the rendering technologies, she will find even static data jarring to watch. The display system must evoke a compelling sense of presence, this involves real-time head tracking and fast rendering of the 3-D scene according to the viewer's head position. In tele-immersion [6], the display used is a spatially augmented display and not a head-mounted display (HMD), and the rendered components are not prestored perfect virtual objects, but real range data acquired online. In addition, these data are transmitted over the network before being displayed. It is important that the rendering speed be higher than the acquisition speed so that user's viewpoint changes have a guaranteed refresh response on the screen.

A technically challenging issue is that, for true bidirectional communication, the capture and display sides of the system must be collocated. Immersive displays currently have low light conditions which make the acquisition of quality images from CCD cameras difficult. Further, the viewer must typically wear polarized or shutter glasses and possibly a head tracking device which does not give him a particularly natural appearance. From the display point of view, inserting many cameras around the display tends to detract from the compelling 3-D percept. Moving cameras to less obtrusive positions causes their viewpoints to be unaccommodating for reconstruction. To date, we have not constructed true duplex telecubicles.

The final and most important question for telepresence systems is "what defines the sense of presence for users?" Can we make a telepresence system that is as effective as "being there"? This is more of a psychological question than a technical one, but understanding factors that evoke the human perception of presence would allow us to focus our technical resources more effectively. The prototypes we describe here offer the first opportunity to perform meaningful psychophysical experiments in order to discover what can make telepresence effective.

B. Related Work

We have divided our discussion of related work into systems and algorithms. The most similar immersive telepresence system is the 3-D video-conferencing system developed under the European 5th Framework Programme's VIRTUE project [7]. Its current version captures a scene with four cameras mounted around a display. After foreground detection and rectification, a block and pixel hierarchic algorithm computes a disparity map while a special depth-segmentation algorithm runs for the user's hands. At the display side, novel views of remote conference participants are synthesized. The next closest as a system is the Coliseum telepresence system [8] which produces synthetic views using five streams based on a variation of the visual hull method [9]. One of the earliest successful efforts is the Virtualized Reality system at Carnegie Mellon University (CMU) [10], [11]. This was first based on multibaseline dense depth map computation on a specialized architecture. More recent versions [12] are based on visual hull computation using silhouette carving. Its commercial version, manufactured by Zaxel, is used in a real-world teleconferencing system [13] overlaying remote participants on HMDs for augmented reality collaboration.

In the systems category, we should also mention the first commercial real-time stereo vision products: the triclops and digiclops by Pointgrey Research, the Small Vision System by Videre Design, Tyzx Inc.'s DeepSea based systems, as well as the Komatsu FZ930 system [14]. These systems are not, however, associated with any telepresence application. With respect to multicamera systems, similar to the latest version of our tele-immersion system, we refer the reader to CMU's newest 3-D room [12] as well as to the view-dependent visual hull system at MIT [9], the Keck laboratory at the University of Maryland [15], and the Argus system at Duke University [16].

Stereo vision has a very long tradition and the interest in fast and dense depth maps has increased recently due to the ease of acquiring video-rate stereo sequences with inexpensive cameras. A recent paper by Scharstein and Szeliski [17] provides an excellent taxonomy of binocular vision systems and a systematic comparative evaluation on benchmark image pairs. Further evaluations with emphasis on matching metrics and discontinuities, respectively, can be found in [18]–[20]. Among the area-based correlation approaches, the most closely related to our system is Sara's work [21], the classic real-time implementation of trinocular stereo [22], [23], and the recent improvements on correlation stereo by Hirschmuller *et al.* [24]. Regarding trinocular stereo vision systems, we refer to the recently reported trinocular systems based on dynamic programming in [25] and [26].

C. Challenges in 3-D Acquisition Through Stereo

In the last section, we described the systems challenges for immersive telepresence. Here we focus on the methodological challenges in the problem we address: environment scanning using multiple cameras. It is important to note that there are also active methods of scanning using laser cameras or systems enhanced with structured light. Laser cameras are still very expensive and structured light systems are still immature for motion and arbitrary texture. Nevertheless, the performance of passive techniques like ours can be improved with projection of unstructured light to complement missing natural texture on surfaces.

When using multiple cameras, we have two choices of the working domain. We can choose either pairs/triples and obtain depth views from each such cluster or we can work volumetrically where the final result is a set of voxel occupancies. While stereo methods rely on matching, volumetric methods can reduce matching to photo-consistency or even just use silhouettes. To date, we have used combinations of pairs/triples in order to guarantee real-time responsiveness and keep the system scalable in the number of depth views. As is widely known, stereo is based on correspondence. The number of possible correspondence assignments without any assumption is exponential. There are two main challenges here: nonexistence of correspondence in case of half-occlusions or specularities and nonuniqueness in case of homogeneous (infinite solutions) or periodic (finite countable solutions) texture.

All existing real-time stereo methods are greedy algorithms which choose the "best" correspondence by considering some finite neighborhood, but they never backtrack to correct a depth value. The match is established by maximizing a correlation metric, with a subsequent selection of the best match. Overly strict selection criteria can result in holes (no valid match) which are larger than areas with no texture or very loose criteria can create multiple outliers (wrong matches). Confidence in our similarity metric is significantly increased if we increase the size of the correlation kernel, but this creates erroneous results at half-occlusions, in particular when one of the two areas in an occlusion does not have significant structure. The most recent prototype resolves some of the outlier problems by using large correlation kernels in a binocular algorithm. Our current system constructs multiple depth views, and we apply strict selection criteria because we anticipate that holes at occlusions will be filled by neighboring depth views.

This merging of views brings us to the problem of registration of multiple depth views to a common coordinate frame. Calibration of many cameras in a large space is a challenging task. When using a reference object for calibration, registration error grows with distance from the reference object. This means that while two depth views are fused correctly when the reconstructed points are close at the location of the reference object during calibration, there is a drift between them when the 3-D points are far from that location. The only remedy for misregistration is a unique volumetric search space, which is part of our ongoing research.

II. Evolution of Tele-Immersion Prototypes

Since our group at the University of Pennsylvania joined the National Tele-immersion Initiative (NTII), we have participated in the development of several tele-immersion prototypes. NTII was conceived by Advanced Network and Services with the expectation that immersive telepresence applications could be a driving application for the capacity of Internet2.

The first working networked tele-immersion prototype (telecubicle) that computed 3-D stereo reconstructions, transmitted them via TCP/IP, and rendered them in a stereoscopic display was demonstrated in May of 1999. The rig is illustrated in Fig. 1(a) and included a pair of Sony XC77R cameras (center), and a computer monitor (CRT) capable of generating stereo views for shutter glasses. Computation was provided by a pair of Pentium 450 PCs, one driving the stereoscopic display and the other capturing image pairs and generating stereo depth maps. Binocular correlation stereo was used to generate a cloud of depth points, which could optionally be triangulated using Jonathan Shewhuk's Triangle code [27]. A rotated triangulated depth view is illustrated in Fig. 1(b).

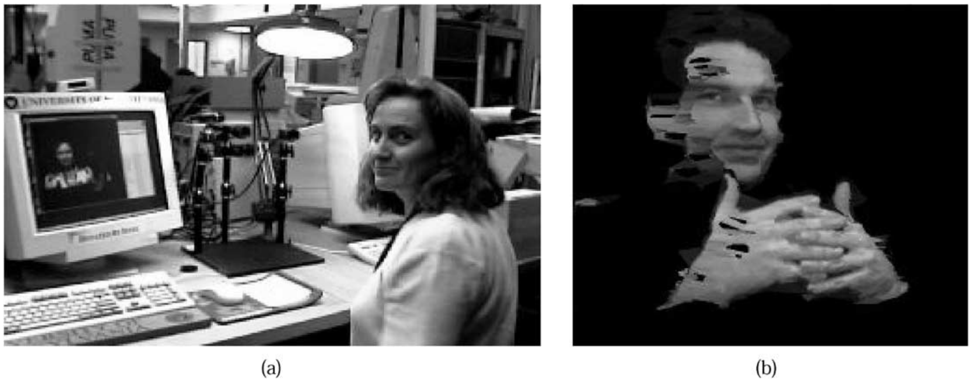
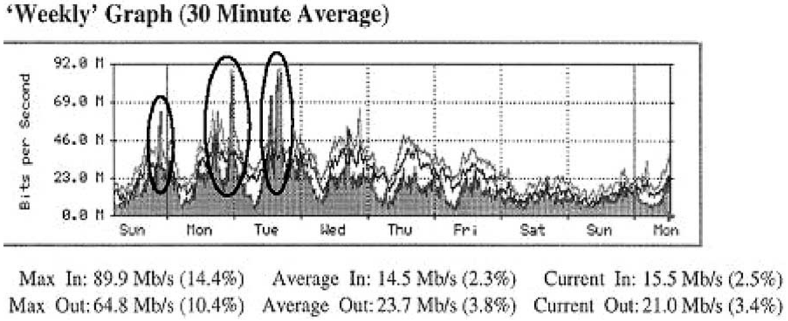


Fig. 1. (a) First networked prototype used shutter glasses and a monitor for stereoscopic viewing. (b) Binocular stereo depth maps were triangulated and rendered.



Max In: 89.9 Mb/s (14.4%) Average In: 14.5 Mb/s (2.3%) Current In: 15.5 Mb/s (2.5%)
Max Out: 64.8 Mb/s (10.4%) Average Out: 23.7 Mb/s (3.8%) Current Out: 21.0 Mb/s (3.4%)

Fig. 2. Traffic peaks into the research triangle on May 2000 demo dates.

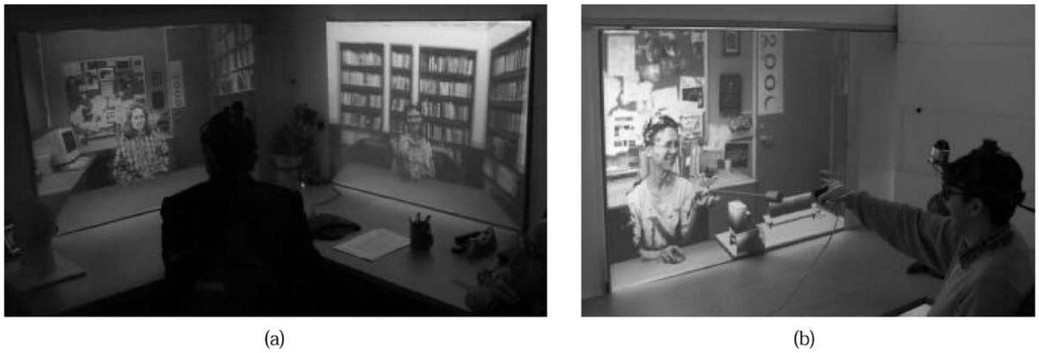


Fig. 3. Working prototypes from May and October 2000. (a) In May, the University of Pennsylvania and advanced network and services each transmitted five simultaneous trinocular depth streams to the University of North Carolina Chapel Hill. (b) In October, 3-D interaction and synthetic objects were added to the collaborative environment.

The next milestone in the tele-immersion project occurred in May 2000 at a major collaborative demonstration among Advanced Network and Services in Armonk, New York, the University of North Carolina (UNC), Chapel Hill, and the GRASP Lab at the University of Pennsylvania. The major innovations for stereo environment scanning were a change from binocular to trinocular correspondence, using a suite of seven Sony DFW-V500 1394 cameras combined in overlapping triples, background subtraction and parallelization of the stereo implementation for quad-processor Dell servers. Naturally, the shear volume of computation and data for transmission was greatly increased. We achieved our goal of driving Internet2 bandwidth as demonstrated by the plot of traffic into the Research Triangle on the dates of our demo and rehearsals (see Fig. 2).

The demo scenario involved transmission of five simultaneous 3-D views per temporal frame from both Advanced and GRASP. These were combined in the sophisticated immersive display provided by UNC. This included two large display screens, one for Armonk and one for Philadelphia, each with a pair of projectors providing differently polarized left–right stereoscopic views. A High-Ball [28] head tracker provided the viewer's head position so 3-D views could be rerendered correctly for the user's viewpoint. The display is illustrated in Fig. 3(a). The camera rig is illustrated in Fig. 5(a) and an example temporal frame is in Fig. 4. The cameras were arranged at uniform height in an arc in front of the user. Although the correlation metric remained the same, new methods for rectifying the triple as two independent pairs and combining the correlation scores for the current depth estimate were

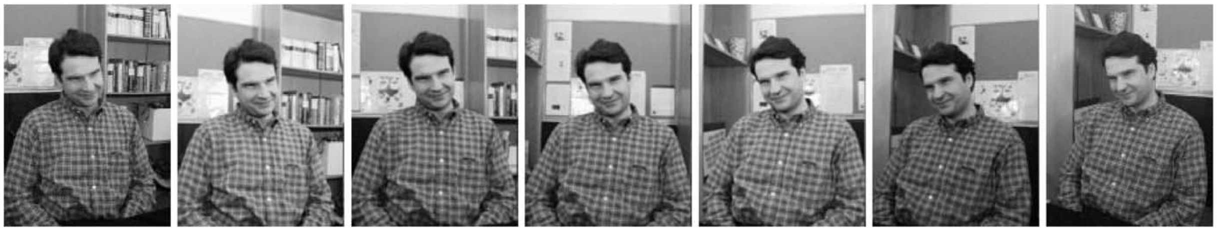


Fig. 4. Seven camera views.

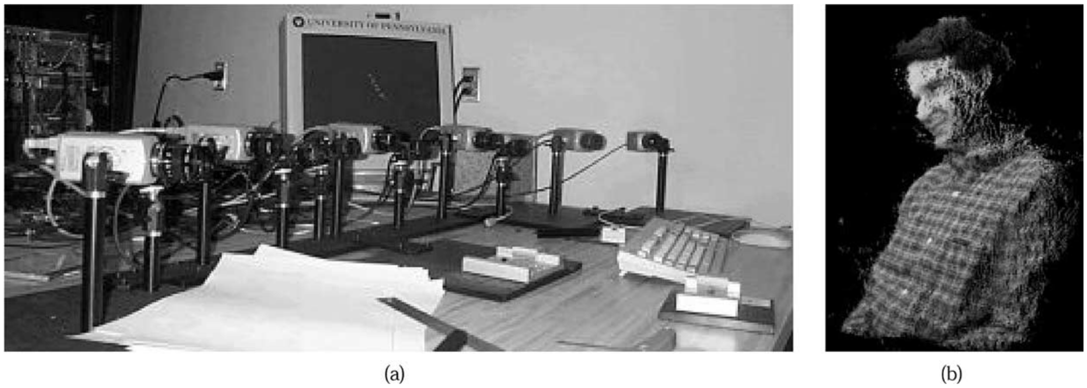


Fig. 5. Seven-camera rig and combined rendered 3-D views.

TABLE I
PERFORMANCE STATISTICS FOR VERSIONS OF THE TELE-IMMERSION SYSTEM

Milestone	Views	Image Dims	Disparities	Kernel	#CPUs	Proc. Time
May '99	Bino	240 × 320	32	5x5	1-P450	1000ms
May '00	Trino	320 × 240	64	5x5	4-P550	450ms
Oct '00	Trino	320 × 240	64	5x5	4-P550	350ms
Aug '02	Trino	320 × 240	64	5x5	2-P2.4	125ms
Nov '02	Bino	640 × 480	100	31x31	450-Alpha	125ms

introduced [29]. A combined rendering of the depth views generated from the images in Fig. 4 is illustrated in Fig. 5(b).

In October of 2000, the next important augmentation of the tele-immersion prototype was demonstrated. Van Dam and his colleagues from Brown University integrated 3-D interaction with the UNC display system [30]. Users could use a magnetic pointer to manipulate and create synthetic objects in the shared 3-D space. This scenario is illustrated in Fig. 3(b). The stereo system, with which we are mainly concerned in this paper, remained largely the same except that a simple prediction scheme was added where the disparity search was limited to a restricted range centered about the computed disparity for the last frame. Although this seems like a minor alteration, it had a significant impact on the cycle time of the stereo calculation (see Table I). By just updating the hardware, the same system was running at 8 fps in August 2002.

The latest collaboration with the UNC and the PSC, funded by the National Science Foundation (NSF), boosted our computation capabilities and gave us the opportunity to be the first to try a room-size real-time reconstruction. The associated increase in the number of input streams as well as in disparity range was a real computational challenge. To make computations as parallel as possible, we temporarily returned to the binocular version. However, this increased the presence of outliers and, thus, the kernel size was drastically increased from 5 × 5 to 31 × 31. The effects of kernel size on reconstruction are discussed in Section IV-B.

In November 2002, we achieved a real-time demonstration of the full cycle at the Supercomputing Conference 2002 in Baltimore. The terascale computing system at the PSC is an HP Alphaserver cluster comprising 750 four-processor compute nodes. Since the PSC is at a remote location, we established one of the first applications where sensing, computation, and display are at three different sites but coupled in real time. To tackle the transmission constraints, an initial implementation contains a video server transmitting TCP/IP video streams and a reliable UDP transmission of the depth maps from the computation to the display site as shown in Fig. 6. This reliable UDP transmission is implemented by a protocol specifically designed for this application, called RUDP. This protocol was designed by the UNC group and provides reliable data transmission required by the application without any congestion control, thereby providing better throughput than TCP. The camera cluster and a snapshot of the rendered scene are shown in Fig. 7.

The massively parallel architecture operates on images four times the size of those used in previous systems (640×480 versus 320×240). Increasing the correlation window size from 5×5 to 31×31 increased computation approximately 36 times. However, we used binocular instead of trinocular stereo so the overhead of matching has been reduced. Overall, the new system requires at least 72 times more computation. Since we do not perform background subtraction, it also adds a factor of 3–4 in complexity.

The correlation window size is the main parameter affecting performance. We ran a series of tests to verify the performance and the scalability of the system. The performance of the real-time system with networked input of video and network output of 3-D streams is constrained by many external factors which could cause a bottleneck. Hence, for performance

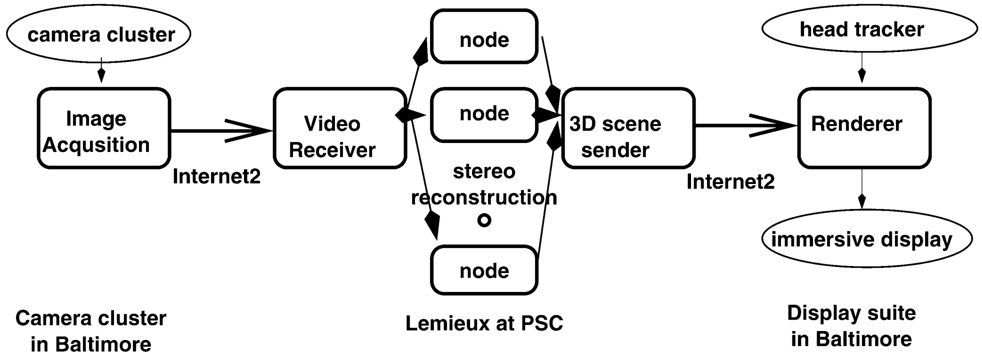


Fig. 6. Images are acquired with a cluster of IEEE-1394 cameras, processed by a a computational engine to provide the 3-D description, transmitted, and displayed immersively.

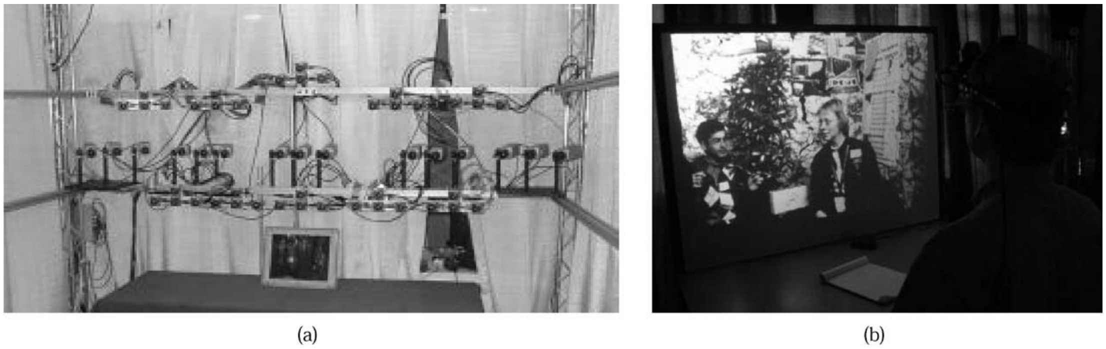


Fig. 7. 30 of the 55 displayed cameras are used in (a) the November 2002 supercomputing conference demonstration and (b) the acquired scene computed at PSC is displayed immersively.

analysis of the parallel algorithm, we switched to file-based i/o. The image streams were read from a disk and we measured the time for image distribution on the cluster, image analysis, and 3-D data gathering from various cluster nodes which contribute to total processing time.

The reconstruction algorithm broadcasts the image to be processed on a particular node in its entirety. Hence, as the number of nodes used for the particular stream increases, so does the broadcast time, as seen in Fig. 8(a). Each processor performs stereo matching on a small strip of the entire image. This is the lowest level of parallelization. The greater the number of processors, the fewer the number of pixels each processor processes. Fig. 8(b) shows the speedup obtained for the "process frame" routine which performs image rectification, stereo matching, and the reconstruction of the 3-D points. We show the processing time for seven different correlation window sizes. The reconstructed 3-D points have to be re-assembled as different parts of the images are reconstructed on different nodes. This gather operation speeds up with number of processors used, due to the smaller amount of data to be gathered from each node.

Based on the above studies, we have observed that the algorithm scales very efficiently with an increasing number of processors per stream. The program is parallelized in such a way so that all streams are synchronized when acquiring the next frame, but each runs independently thereafter to process its assigned image region. Hence, individual processor performance is unaffected. Each stream of images has the same parameters and, thus, execution time is almost the same.

Fig. 9 shows the bandwidth usage for the run during the Bandwidth Challenge at SC2002. The frame rate of up to 8 fps and the data rate over 500 Mb/s observed were achieved with image transmission over TCP. 1080 processors operating on nine binocular streams (120 per stream) were employed for the stereo reconstruction at PSC.

When using multiple cameras on a dynamic scene, synchronization has to be addressed on multiple levels. At the level of camera shuttering, it is solved by triggering the cameras from the parallel port of a server machine. The output of the IEEE-1394 cameras is synchronized with a special box from Point Grey Research and time-stamped so that both subsequent levels of computing and display can be synchronized in software.

III. Stereo-Based Environment Scanning

The stereo algorithm we use is a classic area-based correlation approach. These methods compute dense 3-D information, which allows extraction of higher order surface descriptions. Our system has evolved over several generations as described in Section II, but it continues to operate on a static set of cameras which are fixed and strongly calibrated.

The current versions use clusters of IEEE-1394 cameras combined in triads for the calculation of trinocular or binocular stereo. Computation is parallelized so that bands or even scanlines of the images are processed independently on multiprocessor systems. The general parallel structure of the

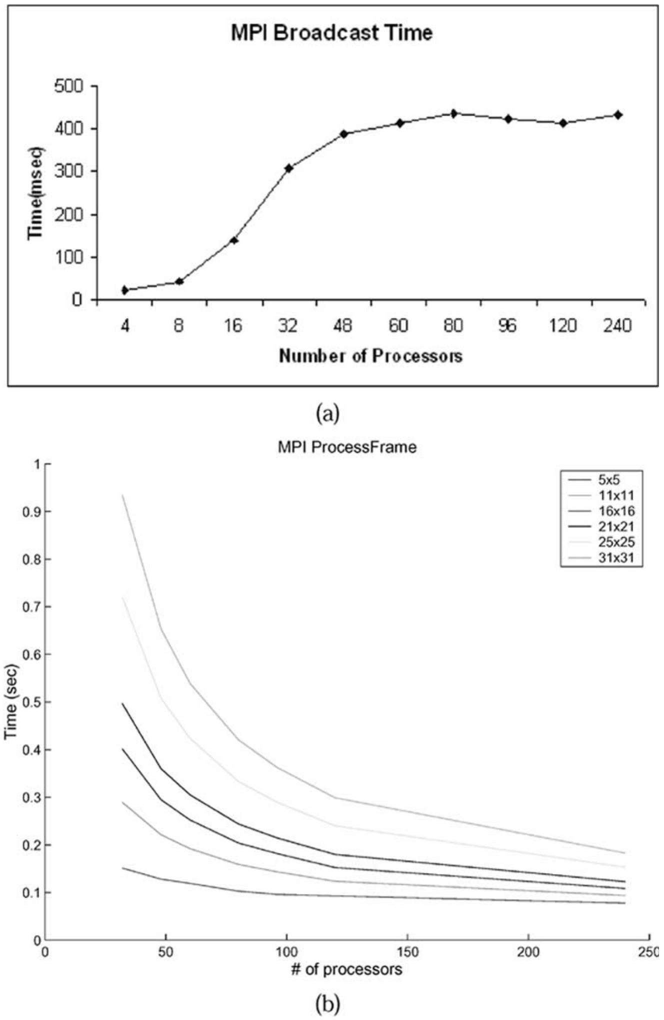


Fig. 8. (a) The time (ms) required to broadcast images to each node increases as the number of processors increases. (b) Total processing time (s) versus number of processors. Each plot corresponds to a different kernel size.

system is illustrated in Fig. 10. All of the images are grabbed simultaneously to facilitate combination of the resulting depth maps at the display side. Each processor rectifies, possibly background subtracts, matches, and reconstructs points in its corresponding image band. When all processors associated with a depth view have finished processing, the texture and depth map are transmitted to a remote renderer. The depth is encoded as two or three unsigned color image planes of texture, plus one unsigned short image plane where $1/z$ values have been scaled into unsigned short. Background subtraction has been omitted in the most recent versions of the prototype. It may have a role to play, however, in computationally less powerful settings, allowing full reconstruction of static background to be calculated only occasionally.

A. Binocular 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 focused 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 overall superior depth maps. In the end, we concluded that the quality of the depth maps was more important to our system and, thus, all of our prototype systems use MNCC. The reconstruction algorithm begins by grabbing images from two or three strongly calibrated cameras. The system rectifies the images so that their epipolar lines lie along the horizontal image rows to simplify the correspondence search.

The MNCC metric has the form

$$corr_{\mathrm{MNCC}}(I_L, I_R) = \frac{2\,\mathrm{cov}(I_L, I_R)}{\sigma^2(I_L) + \sigma^2(I_R)} \tag{1}$$

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 metric above produces a correlation profile $c(u, v, d)$ where disparity d ranges over acceptable integer values. Selected matches are maxima (for MNCC) in this profile.

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 [31]. A hypothesis is correct if the epipolar lines for the original point $[u, v]$ 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 [32]–[35]. Our May 2000 telecubicle configuration, illustrated in Fig. 5(a), was designed to "surround" the user with cameras. The logistics of synchronously capturing and transmitting IEEE-1394 images among multiple computer servers forced us to limit the number of cameras while attempting to cover as much of the scene as possible by arranging the cameras in a single arc and using overlapping triples (as opposed to common "L"-shaped camera arrangements). This configuration 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. For example, in an L-shaped configuration with a central reference camera, the upper image can be rectified to column-align with the reference image while simultaneously the right image is row-aligned. This makes the process of testing correspondences in the third (upper) image much simpler than the lookup table (LUT) and linear approximation schemes we describe below.

Following Okutomi and Kanade's observation [36], 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 $\langle C_R, R\rangle$. In general, 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 [29], 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, and then the computed correlation for each tested $[u_R, v_R, d_R]$ was added to that

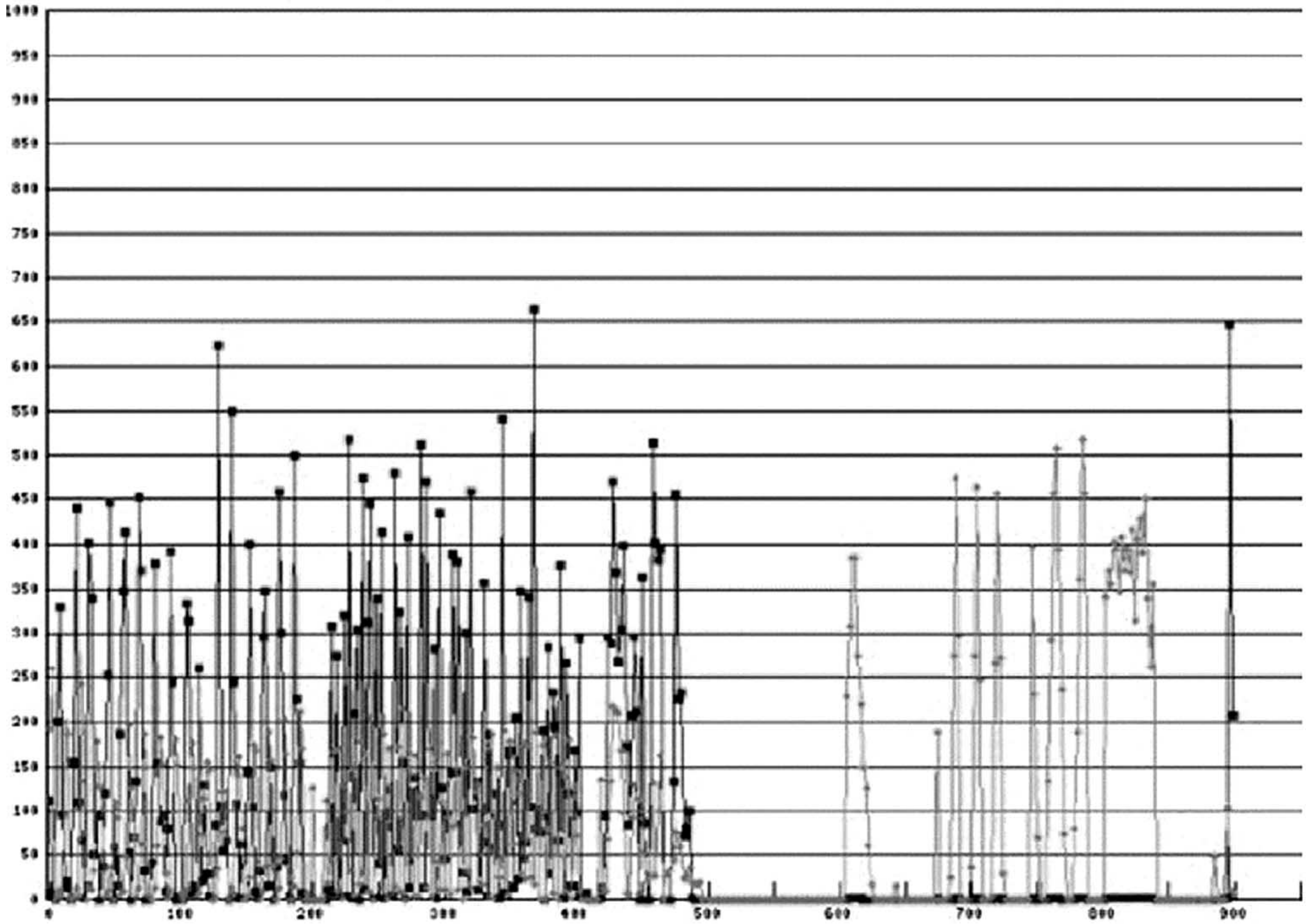


Fig. 9. Bandwidth for tele-immersion usage on November 19, 2002, at the supercomputing conference.

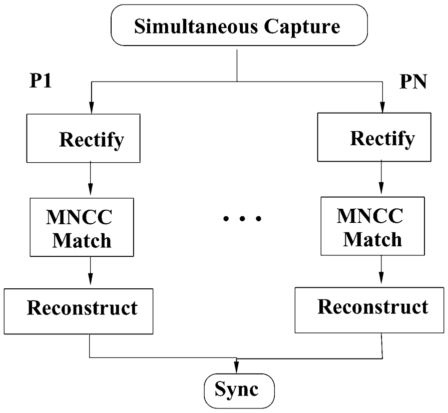


Fig. 10. Parallelized system.

precomputed for the corresponding $[u_L, v_L, d_L]$. This results in large correlation LUTs for the left image pair.

The second method was an attempt to avoid large LUTs 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 LUT 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 foreground 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 an LUT of the location in C_L corresponding the current pixel in C_R (based on the right–left rectification relationship). 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 $\mathrm{corr}_L(u_{C_L}, v_{C_L}, \widehat{d_L})$. We select the disparity d_R which optimizes

$$\mathrm{corr}_T = \mathrm{corr}_L(u_{C_L}, v_{C_L}, \widehat{d_L}) + \mathrm{corr}_R(u_{C_R}, v_{C_R}, d_R).$$

The algorithm has been described in detail in [3] and [37].

IV. Experimental Evaluation

The speed of our systems is relatively straightforward to measure (see Table I), although the causes of lags in captured camera frames or network transmission are not always easy to deduce. We have frequently referred to the quality of reconstructions as the second essential factor for the compelling sense of presence demanded in tele-immersion systems. The accuracy of stereo reconstructions is much more difficult to measure. In previous work [2], [3], we have used registered laser range data and stereo

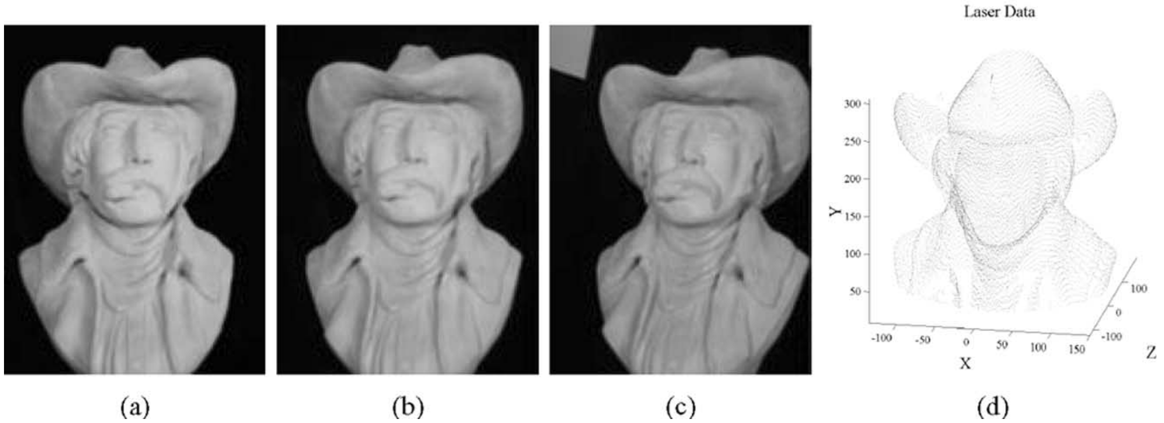


Fig. 11. (a)–(c) Trinocular image data registered to (b) laser data.

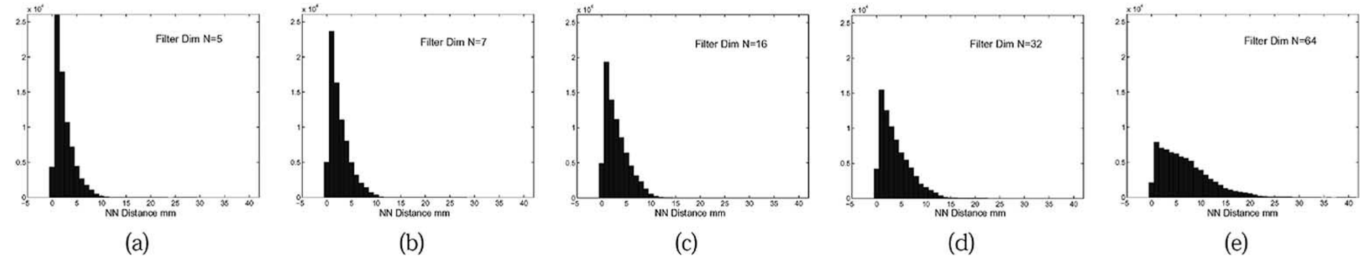


Fig. 12. Smoothing of the NN error histogram with increasing mask size ($N \times N$).

reconstructions of the same object (Fig. 11) to evaluate the accuracy of our systems.

The two most significant concerns regarding the quality of output of the wide-area stereo algorithm are identified to be the accuracy of disparity computation at depth discontinuities, and the registration errors observed when merging reconstruction results from multiple stereo pairs or triplets.

A. Global Effects of Kernel Size

Most recently we have examined the consequences of the large mask sizes used in the current prototype on the accuracy of our reconstructions. Using our registered dataset, a nearest neighbor (NN) error metric (each reconstructed point is assigned an error equal to the distance to the nearest registered laser data point) was computed for reconstructions using mask sizes of 5×5, 7×7, 16×16, 32×32 and 64×64. Fig. 12 illustrates the diffusing effect of increasing mask dimension N on the error histogram. We can see the effect of large masks on the reconstructed data in Fig. 13. The disparity map is much smoother for $N = 32$, and outliers are significantly reduced, but there is also a change in shape from Fig. 13(c) to (d). We can also observe the flattened halo (boundary overreach) at the boundary of Fig. 13(b), caused by smoothing across the occlusion boundary.

When selecting kernel size, we have observed that we obtain a more pleasing result for large-size kernels than smaller ones. That is, end users are less disturbed by a result that includes minor distortion and artifacts at occlusion boundaries than a result that includes holes and outliers. A reason for this is that outliers look like dust or confetti floating in the scene and prevent an observer from clearly perceiving the form of reconstructed objects.

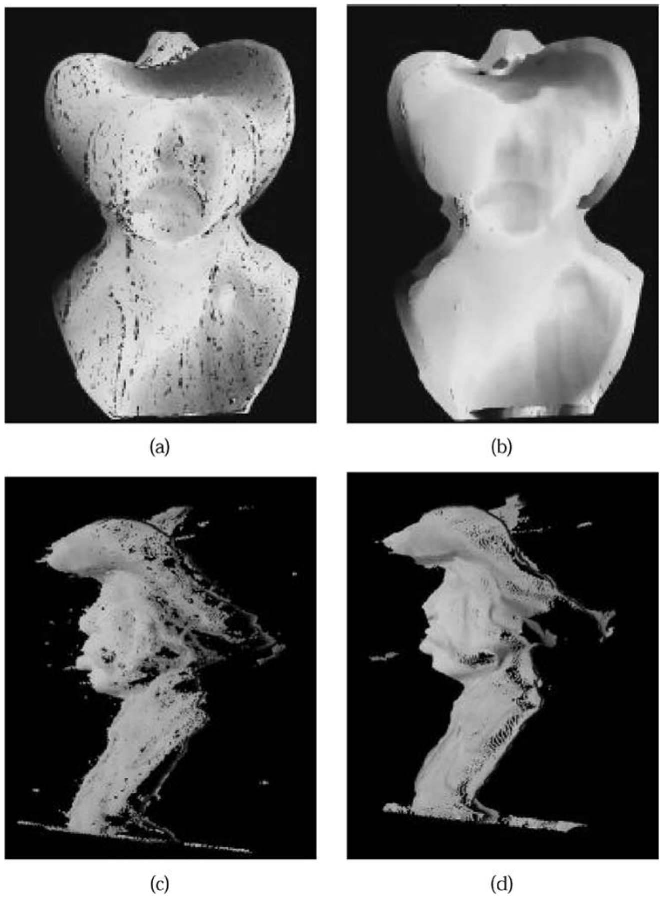


Fig. 13. Disparity maps and rotated reconstructions for $N = 7$ and $N = 32$.

B. Effects of Kernel Size on Discontinuities

Inaccuracies in the disparity map computed by the algorithm are mainly observed at image regions corresponding to depth

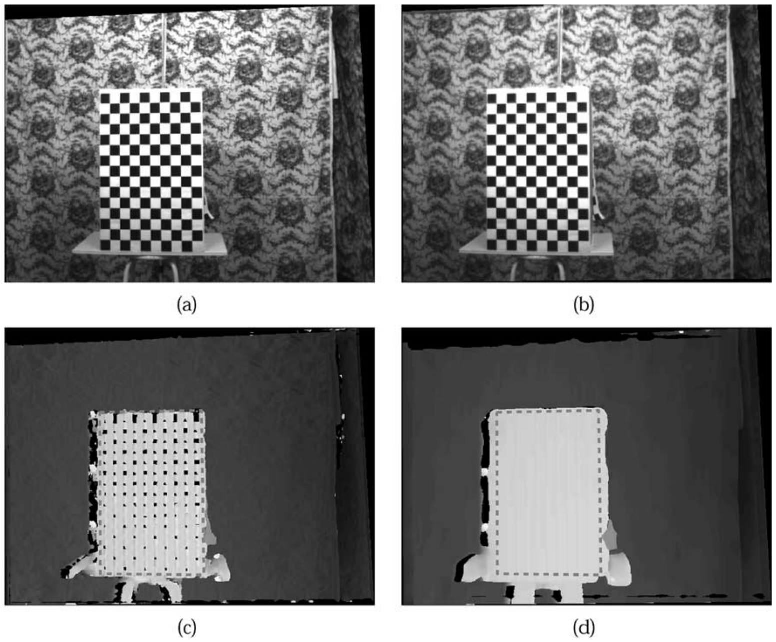


Fig. 14. (a), (b) Rectified stereo pair and derived disparity maps for kernel sizes (c) 11 and (d) 31. In the disparity maps, the dashed line marks the location of the depth discontinuity due to the foreground surface and was manually estimated.

discontinuities. This inaccuracy depends on the size of the kernel. When a small kernel size is used, erroneous pixel correspondences introduce holes and noise in the disparity image and, consequently, gaps and outliers in the reconstruction. In contrast, when the kernel size is increased, more pixel correspondences are established and the disparity image is smoother at depth discontinuities, however the estimated disparity is usually inaccurate in such image regions. More specifically, a spurious border is observed around such discontinuities, a phenomenon that is referred to as "boundary overreach" in the literature. The width of this border is approximately half the diameter of the correlation mask. The error of the depth estimation within this border is quite systematic: typically, the depth of the foreground surface (of the depth discontinuity) is assigned to the pixels within this border, while these pixels arise from the background surface.

In Fig. 14, the above cases are shown through the reconstruction of a simple scene. This scene contains a flat, checkerboard-textured surface, at a frontoparallel posture relative to the cameras. The size of each square was 50×50 mm^2. In Fig. 14(d) the boundary overreach phenomenon is illustrated by marking the true location of the discontinuity. It can be clearly seen that the border (seen around the dashed lines) is assigned disparity values that correspond to the depth of the foreground surface, while these pixels belong to the background.

One way to filter pixels that are included in such artifacts is to perform a "left–right consistency" (LRC) check [38], that is, to require that $d = \mathrm{disp}(I_r(x, y))$ is (approximately) equal to $d' = -\mathrm{disp}(I_l(x + d, y))$ and vice versa (for d and d'), where I_l, I_r is the stereo pair and $\mathrm{disp}(x, y)$ is the estimated disparity value at at image coordinates x, y. The disparity profiles of Fig. 15 illustrate the behavior of the algorithm near discontinuities for the same test object as in Fig. 14. Small kernel sizes increase the occurrence of holes and outliers, while larger kernels increase the boundary overreach. Last, we observe that the LRC filtering method exhibits an over-censoring behavior, as it rejects more than the erroneously reconstructed pixels [Fig. 15(b) and (d)]. In this particular example, it is clear that pixels within $x \in [167, 173]$ have been wrongly rejected, since the whole foreground surface is fully visible to both cameras and, thus, the miscorrespondence of these pixels does not originate from an occlusion (see Fig. 14).

Another approach to refining the disparity map is to apply a median filter to suppress outliers, which typically occur as high spatial frequency noise. In Fig. 16, different modes of operation of the algorithm presented above are demonstrated.

The introduction of the LRC filtering method does not double the computational cost of the algorithm, because when estimating the disparities in one direction (e.g., from left to right) the kernel correlations are recorded and utilized when scanning in the opposite direction. This can be verified by the measurements presented in Table II, where the execution times of the standard mode of operation and the one including LRC are compared. To obtain these measurements, the execution times on each processor were independently recorded over 20 frames of input and then averaged.

The output of the LRC filtering process provides a more conservative result in terms of characterizing reconstructed pixels as erroneous, however, it drastically reduces spurious matches in the output. In particular, the matches eliminated in these regions include background points that are occluded in one image and visible in the other. There is no true match for these points in the occluded view, so correspondences tend to be different in the two images and hence filtered out by LRC. In addition, the LRC method filters spurious matches due to a lack of texture or texture periodicity. It ought to be noted that other approaches to the boundary overreach problem exist in the literature [24], [39]. Their parallel implementation is not as straightforward as

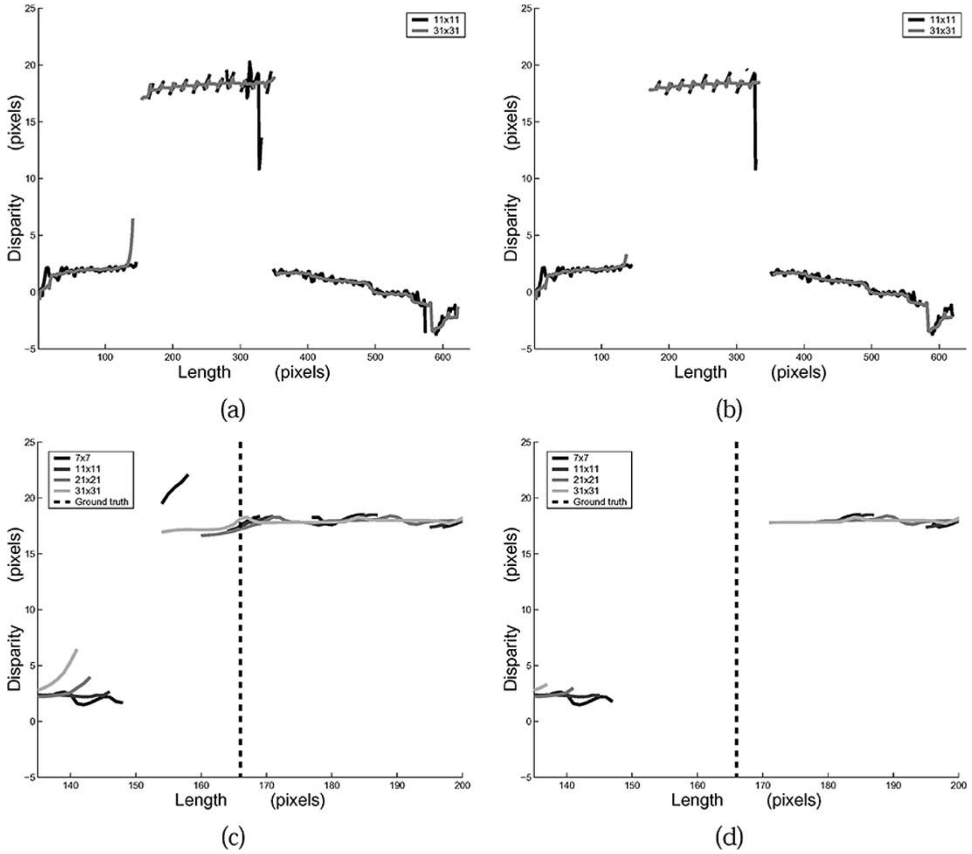


Fig. 15. Disparity profiles across a scanline (200/480) of the stereo pair shown in Fig. 14. The top graphs show the profile along the entire scanline, while the bottom ones focus on the discontinuity, which occurs at $x \approx 167$. In these latter graphs, the vertical dashed line marks the true location of the discontinuity in the image, which was manually estimated. In the right column, graphs were constructed after applying the LRC filtering method. All disparities are presented with respect to the right image of the stereo pair. Graph legends indicate kernel sizes.

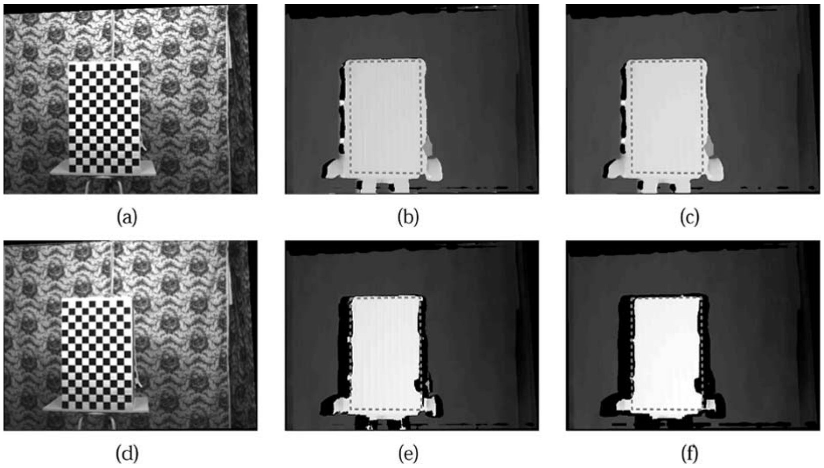


Fig. 16. Stereo pair and generated disparity maps demonstrating the effects of LRC and median filtering. Columns, from left to right: (a), (d) stereo pair, (b) raw algorithm output and (e) application of LRC, and (c) application of median filter and (f) application of LRC and median filter (bottom). The median filtering mask was 7 × 7 pixels, and disparities are presented with respect to the right image of the stereo pair. The dashed lines are the same as in Fig. 14.

that of LRC and their evaluation and potential integration into our system is included in future plans. Finally, the introduction of the median operator significantly suppresses the presence of outliers in the result and can be used to fill minor holes in the disparity image.

C. Registration Error

When combining reconstruction results obtained from multiple stereo image pairs, a "registration error" can be observed. This error refers to the fact that independent, but calibrated and registered, reconstructions of the same world point are not well aligned. It is mainly encountered as the variance of the z coordinate of different reconstructions of a world point. The graphs in Fig. 17 illustrate the problem. They were obtained by plotting the data from the following experiment: six cameras were used to create three binocular views. Views were equidistant and loosely aligned, with each one yielding an independent reconstruction of a commonly observed scene. The scene contained

TABLE II
MEASUREMENTS OF EXECUTION TIME ON A SUPERCOMPUTER, AT PSC, SHOWING THAT THE APPLICATION OF LRC FILTERING IMPOSES A VERY SMALL ADDITIONAL COMPUTATIONAL COST. THE TABULATED VALUES REFER TO THE MEAN AVERAGE EXECUTION TIME REQUIRED TO PROCESS ONE FRAME OF INPUT. UNIT TIME IN THE SECOND AND THIRD COLUMN IS IN SECONDS AND IN THE FOURTH COLUMN IS IN MILLISECONDS

Processor #	LRC	No LRC	Difference
8	50.781251	50.267734	514
16	25.315330	25.288857	265
32	12.730962	12.666340	64
40	10.202700	10.151524	51
60	6.850314	6.814646	35
80	5.182481	5.152075	30
120	3.495402	3.485919	9
160	2.678267	2.773337	4
240	1.849374	1.849731	3
480	1.270215	1.197713	7

the test object of Fig. 14 which was placed at a total of six different positions, thus yielding equal variants of the scene, which were all reconstructed. The surface was always positioned so that it was approximately frontoparallel to the central view and translated at different distances along the direction normal to the baseline of the central view (the Z axis). Camera calibration was performed using the Matlab Calibration Toolbox by Bouguet. In the graphs, the reconstructed points across 10 mm of the X axis are plotted for the three views, with each graph corresponding to a different position of the test object. Points at the top left of each graph represent points on the background surface, which was approximately stable across the different variants of the scene. The points at the bottom right represent points on the object surface.

A general observation is that the registration error is proportional to the distance from the cameras, producing the systematic ordering of world points across depth illustrated in the example. The usual predicted error in depth is proportional to the depth of the point squared, which would cause reconstructions of the same world points to be randomly positioned, if the predicted error were the only component of the observed misregistration. Thus, a component of misregistration is calibration error as well. As we also mention in the conclusion, misregistration can be alleviated if we use a common search range like a volumetric domain for all views. In this case, calibration errors do not show as double phantoms but as inaccuracies in the voxel level.

V. PREDICTION AND MODELING

In less compute-intensive settings, a possible approach to improving temporal performance for stereo is to exploit the fact that scenes do not change radically from frame to frame. The simplest form of prediction, based on the temporal coherence of our image streams, is to assume that nothing changes. In other words, predict that the last observed disparity will be the next observed disparity. Our current prototypes limit their disparity search at each timestep to a fixed range about the last observed disparity. This breaks down when the subject moves in front of a distant background, but it is a powerful assumption for all but the boundary pixels about the subject. Occlusion boundary issues in dense stereo tend to make these pixels more problematic, independent of motion.

Once again addressing the tradeoff of speed and accuracy, can we exploit temporal coherence to a greater degree using more sophisticated motion-based prediction techniques? The cost of computing dense optical flow is high, so can we apply it judiciously and still improve speed without sacrificing quality?

A. *Predicting Disparity Windows*

We base our approach on the assumption that there are smooth surface patches in the scene that will move coherently as people or objects in the environment move. To identify these surface patches we use a simple flood-fill technique to segment the disparity map into regions of similar disparity. Disparities within a region are either limited to fall within a small fixed range of the original seed pixel ($d_{\text{seed}} - t \leq d_{\text{cur}} \leq d_{\text{seed}} + t$), or a disparity gradient limit is applied so that the region is grown until all of the boundary pixels exceed this limit ($|d_{\text{prev}} - d_{\text{cur}}| < \Delta_{\text{lim}}$). The segmented regions are represented by their bounding upper-left, lower-right image locations and thus are effectively treated as overlapping rectangular windows.

In order to predict the location of a particular window at the next time step, we use a single optical flow calculation per window in the left (reference) image sequence to estimate its motion. We can use the disparity range for each window to locate its corresponding window in the right (nonreference) image and calculate the image motion for the right image. This allows us to predict a disparity range and image location for the surface represented by the disparity window at the next timestep. We can then perform a region-based correlation on a limited disparity range for each predicted window location.

Our method for integrating disparity segmentation and optical flow for disparity prediction 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) Go to Step 2).

1) Flood-Fill Segmentation: It is more common to use flow fields to provide coarse segmentation than to use similar disparity [40], [41], but our existing stereo system provides dense disparity maps, whereas most fast optical flow techniques provide relatively sparse flow values. Restricting the change in disparity per window essentially divides the underlying surfaces into patches where depth is nearly constant or smoothly varying.

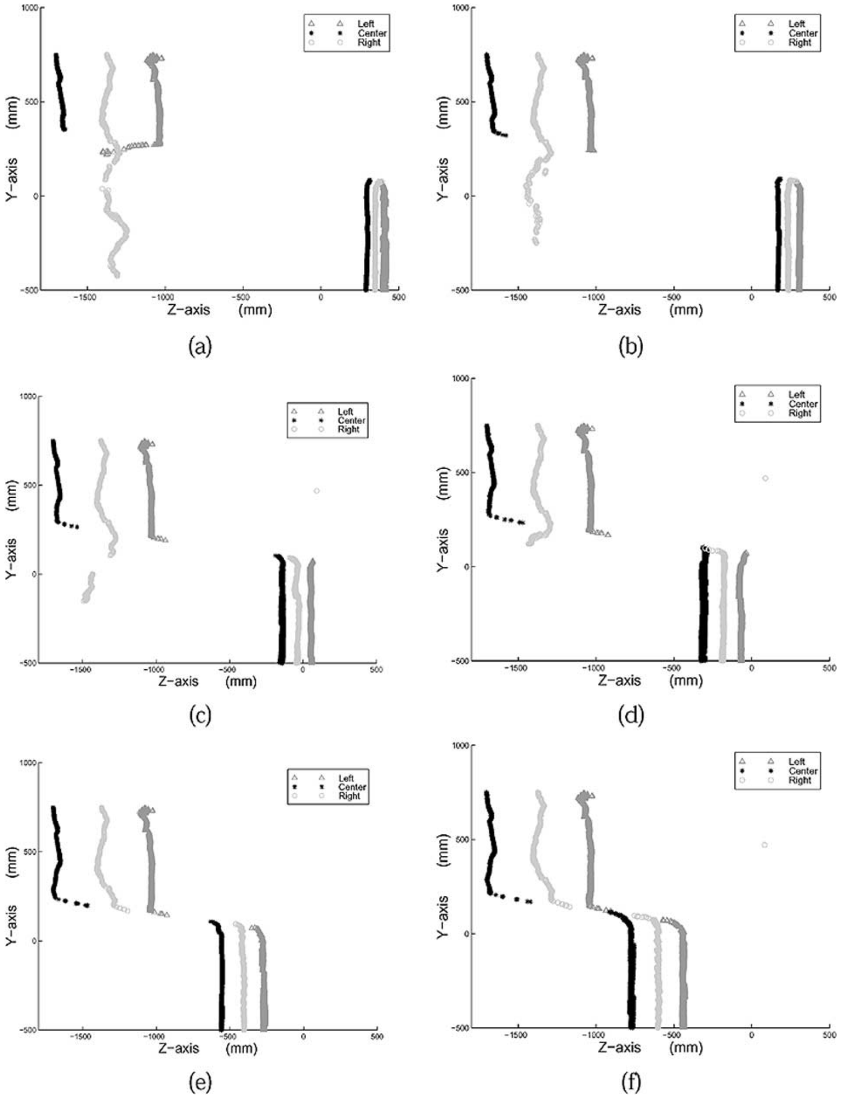


Fig. 17. Registration of reconstructions, simultaneously obtained from three different views. In the graphs, plotted are lateral views of six reconstructions obtained from the motion of a planar foreground surface translating along the Z axis against a, partially occluded, wavy background surface (a curtain). Positive direction of the Z axis is toward the central cameras. In the graphs, pixels reconstructed from the center view are plotted in black, pixels from the left view in dark gray, and pixels from the right view in light gray.

Any efficient region growing method could be applied to cluster the disparities into regions. We have chosen to use flood fill or seed fill [42, pp. 137–141], a simple polygon filling algorithm from computer graphics. We have implemented a scanline version which pops a seed pixel location inside a region 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, nonboundary span on these lines in this range is pushed on the seed stack and the loop is repeated. When the stack is empty the region is filled.

We have modified this process slightly so that the boundary is defined by a condition on the current pixel disparity d_{cur}, with respect to other disparities in the region ($d_{\mathrm{seed}}, \ d_{\mathrm{prev}}$). We start with a mask of valid disparity locations in the disparity image possibly including a background segmentation to eliminate static background pixels. 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 constraint for the region, 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.

The disparity map for pair 59 of our test image sequence is illustrated in Fig. 18, along with the rectified reference image and disparity windows extracted by the flood-fill segmentation using the disparity gradient constraint ($\Delta_{\mathrm{lim}} = 1$). Twenty-two regions were extracted, with mean width disparity range of seven pixels. We maintain only rectangular image windows 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. A window can cover pixels which are not connected to the current region being filled (for example a rectangular bounding box for an "L"-shaped region will cover many pixels

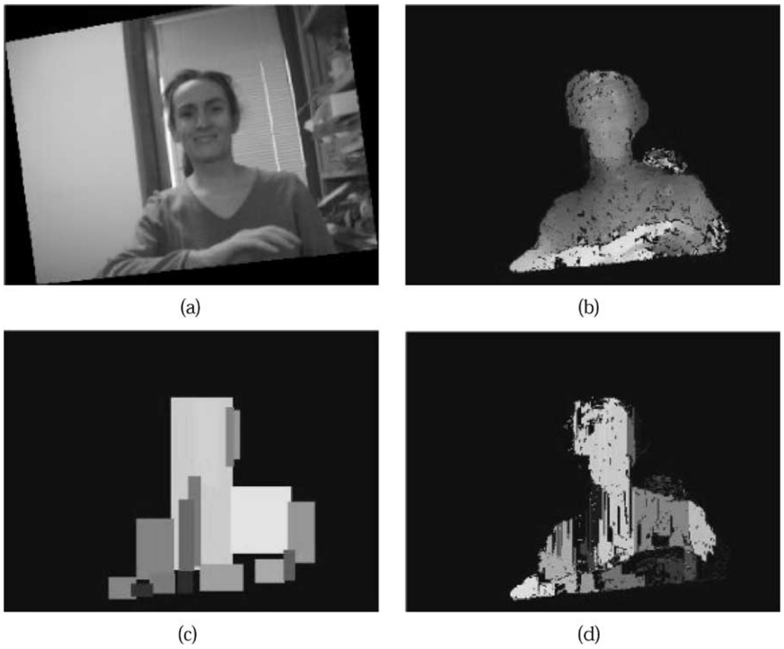


Fig. 18. Disparity windows. (a) Rectified image. (b) Full disparity map. (c) Extracted windows. (d) Pixels labeled with their assigned window.

that are not explicitly in the disparity range) and therefore the windows extracted overlap. This is an advantage when change in disparity signals a depth discontinuity, because if a previously occluded region becomes visible from behind another surface, the region will be tested for both disparity ranges.

As a final step, small regions ($<$ MINREG pixels) are attributed to noise and deleted. Nearby or overlapping windows are merged when the corner locations bounding window W_i expanded by a threshold NEARWIN, fall within window W_j, and the difference between the region mean disparities satisfies

$$\left| \frac{\sum\limits_{R_i} I_D(x_l, y_l)}{N_i} - \frac{\sum\limits_{R_j} I_D(x_k, y_k)}{N_j} \right| < \mathrm{NEARDISP}$$

where R_i and R_j are the set of pixels in two disparity regions, with N_i and N_j elements, respectively.

2) 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 [43], [44] 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. We do not apply an affine flow assumption because of the increased complexity of solving for six parameters rather than just two components of image velocity [45].

For each disparity window, we assume the motion field is constant across the region W_i and calculate a single value for the center pixel. Only one optical flow value is estimated per window. Fig. 19 shows the comparison between flow estimates for 5×5 windows across the full image and values computed for our segmented windows (depicted by the same vector at each window location) for the left image sequence frames 60–64.

For each window represented by its upper left and lower right corner locations $W(t) = [(x_{ul}, y_{ul}), (x_{lr}, y_{lr})]$, we adjust its location according to our estimated flow for the right and left images. We must also adjust the disparity range $D(t) = [d_{\min},\ d_{\max}]$ for each window as follows:

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

Optical flow calculations can sometimes yield poor results, for example, when the subject moves along the depth axis, so we actually expand the windows and disparity range according to the computed motion rather than moving them. Our observation has been that, for normal motion of a subject in the workspace, motion estimates are relatively good.

3) Windowed Correspondence: Window-based correspondence proceeds much as described for the full image, except for the necessary manipulation of windows. Calculation of MNCC using (1) allows overall calculation of the terms $\sigma^2(I_L)$, $\sigma^2(I_R)$, and $\mu(I_L)$ and $\mu(I_R)$ on a once-per-image pair basis. For $\mathrm{cov}(I_L, I_R) = \mu(I_L I_R) - \mu(I_L)\mu(I_R)$, however, $\mu(I_L I_R)$ and the product $\mu(I_L)\mu(I_R)$ must be recalculated for each disparity tested. In the case of our disparity windows, each window can be of arbitrary size, but the disparity range to be checked will be shorter. 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 volume proceeds as described for the full image method.

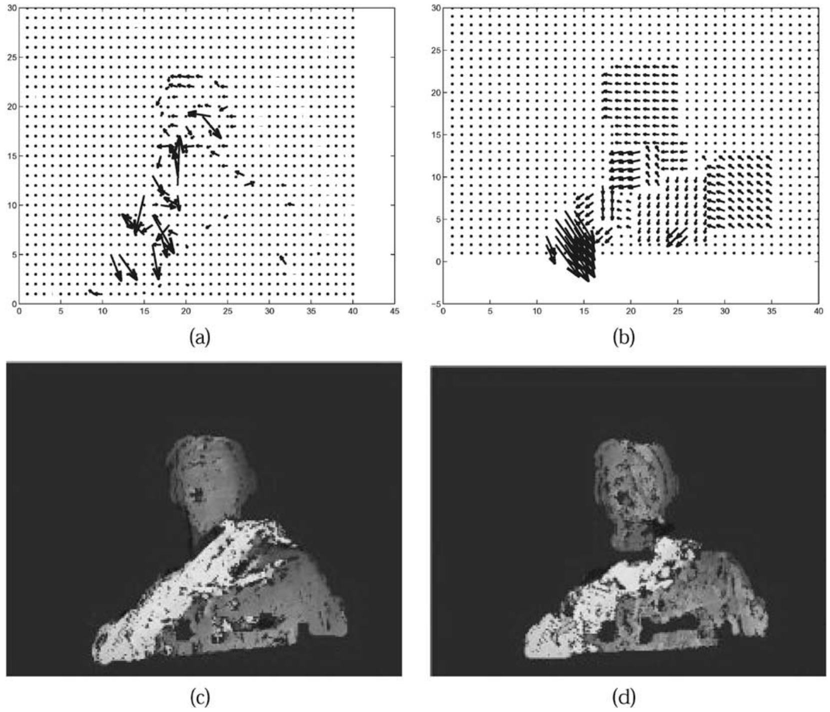


Fig. 19. Flow fields computed for (a) full image and (b) segmented windows (left frames 60–64). (c) Full correspondence disparity map. (d) Regional correspondence disparity map.

B. Computational Complexity

Fig. 19 illustrates the full correspondence versus regional correspondence maps. The regional map is somewhat sparser but it required 42% less calculation to generate. Generally for a reference image with K valid pixels (possibly $M \times N$ or some set of foreground pixels) let us consider the operation of convolving with a mask g of size n_g. We currently do the same per pair calculations $\sigma^2(I_L)$, $\sigma^2(I_R)$, $\mu(I_L)$ and $\mu(I_R)$ for the full and regional matching, so we will discount these in our comparison. The term $\mu(I_L)\mu(I_R)$ over $(d_{\max} - d_{\min})$ disparities requires $Kn_g(d_{\max} - d_{\min})$ multiplications for the full image case, and $\sum_W n_{xi}n_{yi}n_g(d_{\max_i} - d_{\min_i})$ multiplications for the set of extracted windows W, where W_i has dimensions (n_{xi}, n_{yi}). Similarly, calculating $\mu(I_L I_R)$ will require $Kn_g(d_{\max} - d_{\min})$ versus $\sum_W n_{xi}n_{yi}n_g(d_{\max_i} - d_{\min_i})$ multiplications. We have to weigh this savings in the covariance calculation against the smoothing and least-squares calculation per window of the optical flow prediction process.

For temporal estimates over n_t images in a sequence, we have to smooth and calculate derivatives for the images in the sequence in x, y, and t. This calculation requires $(2 \times 3)n_g K n_t$ multiplications for each of two (right and left) image sequences. Solving $Av = b$, using, for example, QR decomposition and back substitution, requires approximately $(12)(n_{xi}n_{yi})$ flops per window.

Finally, for the flood-fill segmentation, each pixel may be visited up to four times (once when considering each of its neighbors), but probably much fewer. The only calculations performed are comparisons to update the window corners and disparity range as well as a running sum of the pixel values in each region. The cost is small compared to full image correlations, so we will disregard it here.

The regional correspondence will be faster if the following comparison is true:

$$(2)Kn_g(d_{\max} - d_{\min}) > (2 \times 2 \times 3)n_g K n_t \\ +(2)\sum_W \left[n_{xi}n_{yi}n_g(d_{\max_i} - d_{\min_i})\right] + \sum_W \left[(12)(n_{xi}n_{yi})\right]. \quad (2)$$

For example, for the disparity frame in Fig. 19(c) and (d), a rough estimate of the relative complexity of a background subtracted full correspondence versus disparity windows combined with background subtraction yields $40\,774\,950 > 23\,531\,823$. This difference on paper was not as striking in the online system, however, we believe that the extracted regions and their motion tell us something useful about the structure of the scene. In the future, we hope to exploit this structure to understand activity in the scene.

VI. Conclusion

We have presented the evolution of a scene acquisition system for tele-immersion. From the beginning, we have focused on near real-time systems which are view-independent so that they can facilitate a rendering speed independent of the acquisition and transmission frame rate and latency. We have also examined techniques to exploit the temporal coherence of stereo sequences, by using prediction to restrict disparity search ranges. Over a number of years we have developed a sequence of binocular and trinocular stereo reconstruction systems, steadily increasing the number of cameras and processors exploited. Today, the most important contribution of our work is wide area acquisition where everything in the scene is captured and thus is "foreground." The resulting increase in number of cameras, input resolution, and disparity range motivated us to use massive parallelization and produce a truly distributed sensing-computing-display system for tele-immersion.

In the immediate future, we are addressing the problems of occlusion and misregistration. In a wide surround distribution of cameras, there is no clear definition of occlusion like the definition of half-occlusion in stereo. Views producing outliers or holes at half-occlusions should be corrected by other views. Such a fusion process necessitates a good confidence metric. To avoid fusion, we will pursue a volumetric approach (see [46] for thorough treatment) where we deal with a unique disparity space for all cameras, but then visibility becomes a problem: which views should be used for photoconsistency or to compute correlation? Parallelization of a volumetric approach becomes a challenging problem with many more interconnections between mutually independent input streams. A volumetric approach would also treat symmetrically the problem of misregistration due to calibration errors by blurring the estimated voxels instead of duplicating depth maps. However, wide-area calibration remains a challenge.

Collocating large immersive displays with cameras for environment scanning in order to provide duplex communication presents further technological challenges. It requires addressing the problem of rendering the scene from viewpoints far from the viewpoints where the input streams were captured for reconstruction. In a collocated display-camera system, it is natural to ask for localization of face and body parts so that head tracking as well as gesture recognition will be accomplished without wearing devices. Full duplex communication will also enable the start of human performance experiments where we will be able to study the question of whether specific collaboration tasks can be better addressed with tele-immersion than with plain or even large-scale videoconferencing.

Acknowledgment

The authors would like to thank J. Lanier, H. Fuchs, and A. van Dam for the wonderful collaboration in the National Tele-immersion Initiative as well as H. Towles and the entire UNC team for their contribution to the November 2002 demonstration. They also thank R. Sara and G. Kamberova for the initial offline implementation of tele-immersion at the University of Pennsylvania.

References

[1] J. Lanier, “Virtually there,” *Scientific Amer.*, pp. 66–75, Apr. 2001.

[2] J. Mulligan, V. Isler, and K. Daniilidis, “Performance evaluation of stereo for tele-presence,” in *Proc. 8th IEEE Int. Conf. Computer Vision (ICCV’01)*, vol. 2, Vancouver, BC, Canada, July 2001, pp. 558–565.

[3] ——, “Trinocular stereo: A real-time algorithm and its evaluation,” *Int. J. Comput. Vis.*, vol. 47, no. 1/2/3, pp. 51–61, 2002.

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

[5] A. Hilton, D. Beresford, T. Gentils, R. Smith, W. Sun, and J. Illingworth, “Whole-body modeling of people from multi-view images to populate virtual worlds,” *Int. J. Comput Graphics*, vol. 16, no. 7, pp. 411–436, 2000.

[6] 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 *Proc. ACM SIGGRAPH*, 1998, pp. 179–188.

[7] P. Kauff and O. Schreer, “An immersive 3-D video-conferencing system using shared virtual team user environments,” in *Proc. ACM Conf. Collaborative Virtual Environments*, 2002.

[8] H. Baker, D. Tanguay, I. Sobel, D. Gelb, M. Gross, W. Culbertson, and T. Malzbender, “The coliseum immersive teleconferencing system,” in *Proc. Int. Workshop Immersive Telepresence*, Juan-les-Pins, France, Dec. 6, 2002.

[9] W. Matusik, C. Buheler, R. Raskar, S. Gortler, and L. McMillan, “Image-based visual hulls,” in *Proc. ACM SIGGRAPH*, 2000, pp. 369–374.

[10] T. Kanade, A. Yoshida, K. Oda, H. Kano, and M. Tanaka, “A stereo engine for video-rate dense depth mapping and its new applications,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, San Fransisco, CA, June 18–20, 1996, pp. 196–202.

[11] P. Narayanan, P. Rander, and T. Kanade, “Constructing virtual worlds using dense stereo,” in *Proc. Int. Conf. Computer Vision*, 1998, pp. 3–10.

[12] G. Cheung, T. Kanade, J. Bouguet, and M. Holler, “A real time system for robust 3-D voxel reconstruction of human motions,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, June 13–15, 2000, pp. 714–720.

[13] M. Billinghurst, A. Cheok, S. Prince, and H. Kato, “Projects in vr: Real world teleconferencing,” *IEEE Comput. Graph. Applicat.*, vol. 22, pp. 11–13, 2002.

[14] T. Naemura, J. Tago, and H. Harashima, “Real-time video based modeling and rendering of 3d scenes,” *IEEE Computer Graph. Applicat.*, vol. 22, pp. 66–73, 2002.

[15] P. Baker and Y. Aloimonos, “Complete calibration of a multi-camera network,” in *Proc. IEEE Workshop Omnidirectional Vision*, Hilton Head Island, SC, June 12, 2000.

[16] D. Brady, R. Stack, S. Feller, L. F. E. Cull, D. Kammeyer, and R. Brady, “Information flow in streaming 3-D video,” in *Three-Dimensional Video and Display Devices and Systems*: SPIE PRESS, 2000, vol. CR76.

[17] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *Int. J. Comput. Vis.*, vol. 47, no. 1/2/3, pp. 7–42, 2002.

[18] J. Banks and P. Corke, “Quantitative evaluation of matching methods and validity measures,” *Int. J. Robot. Res.*, vol. 20, pp. 512–532, 2001.

[19] R. Sara and R. Bajcsy, “On occluding contour artifacts in stereo vision,” in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Puerto Rico, June 17–19, 1997, pp. 852–857.

[20] G. Egnal and R. Wildes, “Detecting binocular half-occlusions: Empirical comparisons of five approaches,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, 2002.

[21] R. Sara, “Finding the largest unambiguous component of stereo matching,” in *Proc. 7th Eur. Conf. Computer Vision*, 2002, pp. 900–914.

[22] O. Faugeras *et al.*, “Real Time Correlation-Based Stereo: Algorithm, Implementation, and Applications,” INRIA, Sophia Antipolis, Tech. Rep. 2013, 1993.

[23] L. Matthies, “Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation,” *Int. J. Comput. Vis.*, vol. 8, pp. 71–91, 1992.

[24] H. Hirschmuller, P. Innocent, and J. Garibaldi, “Real-time correlation based stereo with reduced border errors,” *Int. J. Comput. Vis.*, vol. 47, pp. 229–246, 2002.

[25] M. Agrawal and L. Davis, “Trinocular stereo using shortest path and the ordering constraint,” *Int. J. Comput. Vis.*, vol. 47, pp. 43–50, 2002.

[26] C. Buehler, S. Gortler, M. Cohen, and L. McMillan, “Minimal surfaces for stereo,” in *Proc. 7th Eur. Conf. Computer Vision*, Copenhagen, Denmark, 2002, pp. 885–899.

[27] J. R. Shewchuk, “Triangle: Engineering a 2D quality mesh generator and delaunay triangulator,” in *Applied Computational Geometry: Toward Geometric Engineering*, M. C. Lin and D. Manocha, Eds. Berlin, Germany: Springer-Verlag, 1996, vol. 1148, Lecture Notes in Computer Science, pp. 203–222.

[28] G. Welch and G. Bishop, “Scaat: Incremental tracking with incomplete information,” in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1997, pp. 333–344.

[29] J. Mulligan and K. Daniilidis, “Trinocular stereo for nonparallel configurations,” in *Proc. 15th Int. Conf. Pattern Recognition*, Barcelona, Spain, Sept. 2000, pp. 567–570.

[30] H. Towles, W.-C Chen, R. Yang, S.-U Kum, H. Fuchs, N. Kelshikar, J. Mulligan, K. Daniilidis, L. Bolden, B. Zelesnik, A. Sadagic, and J. Lanier, “3-D tele-collaboration over internet2,” in *Proc. Int. Workshop Immersive Telepresence*, Juan-les-Pins, France, Dec. 6, 2002.

[31] U. Dhond and J. Aggrawal, “Structure from stereo: A review,” *IEEE Trans. Syst., Man, Cybernet.*, vol. 19, pp. 1489–1510, 1989.

[32] Y. Ohta, M. Watanabe, and K. Ikeda, "Improving depth map by right-angled trinocular stereo," in *Proc. 8th Int. Conf.Pattern Recognition (ICPR'86)*, vol. I, Paris, France, Oct. 1986, pp. 519–521.

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

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

[35] D. Murray and J. Little, "Using real-time stereo vision for mobile robot navigation," *Auton. Robots*, vol. 8, no. 2, pp. 161–171, 2000.

[36] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 353–363, Apr. 1993.

[37] J. Mulligan and K. Daniilidis, "View-independent scene acquisition for tele-presence," in *Proc. IEEE and ACM Int. Symp. Augmented Reality*, Munich, Germany, Oct. 2000, pp. 105–108.

[38] P. Fua, "A parallel stereo algorithm that produces dense maps and preserves image features," *Machine Vis. Applicat.*, vol. 6, pp. 35–49, 1993.

[39] M. Okutomi, Y. Katayama, and S. Oka, "A simple stereo algorithm to recover precise object boundaries and smooth surfaces," *Int. J. Comput. Vis.*, vol. 47, no. 1/2/3, pp. 261–273, 2002.

[40] M. Irani, B. Rousso, and S. Peleg, "Computing occluding transparent motions," *Int. J. Comput. Vis.*, vol. 12, no. 1, pp. 5–16, Jan. 1994.

[41] F. Meyer and P. Bouthemy, "Region-based tracking in an image sequence," in *Proc. 2nd Eur. Conf. Computer Vision*, vol. 588, Lecture Notes in Computer Science, G. Sini, Ed., Santa Margherita Ligure, Italy, May 1992, pp. 476–484.

[42] D. F. Rogers, *Procedural Elements for Computer Graphics*, 2nd ed. Boston, MA: WCB/McGraw-Hill, 1998.

[43] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artificial Intelligence (IJCAI'81)*, Vancouver, BC, Canada, 1981, pp. 674–679.

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

[45] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM Comput. Surveys*, vol. 27, no. 3, pp. 433–467, 1995.

[46] G. Slabaugh, B. Culbertson, T. Malzbender, M. Livingston, I. Sobel, M. Stevens, and R. Schafer, "A collection of methods for volumetric reconstruction of visual scenes," Int. J. Comput. Vis., 2003, submitted for publication.

Jane Mulligan received the B.S. degree in computer science from Acadia University and the M.S. and Ph.D. degrees in compuer science from the University of British Columbia, Vancouver, BC, Canada.

She is an Assistant Professor with the Computer Science Department, University of Colorado at Boulder. Prior to joining the University of Colorado at Boulder, she was a Postdoctoral Fellow with the GRASP Laboratory, University of Pennsylvania, Philadelphia. Her current research interests include extracting human models in telepresence settings and environment scanning from mobile robotic platforms.

Xenophon Zabulis received the M.S. degree in computer science and the Ph.D. degree from the University of Crete in 1998 and 2001, respectively.

He is a Postdoctoral Fellow working with Kostas Daniilidis at the Computer and Information Science Department, University of Pennsylvania, Philadelphia, and affiliated with the interdisciplinary GRASP laboratory at the same institution. He is currently involved with multiple-view scene acquisition for tele-immersion. Prior to his current appointment, he was a Postdoctoral Fellow with the Institute for Research in Cognitive Science, University of Pennsylvania, working in the understanding of human binocular vision. His research interests include visual information retrieval by content as well as computational aspects of human and multiple-view vision.

Nikhil Kelshikar received the M.S. degree in computer science from the University of South Florida.

He is a Research Associate working with Kostas Daniilidis at the Computer and Information Science Department, University of Pennsylvania, Philadelphia and is affiliated with the GRASP Laboratory. He is currently working on multicamera reconstruction for tele-immersion.

Kostas Daniilidis (S'90–M'92) received the M.S. degree in electrical engineering from the National Technical University of Athens, Athens, Greece, in 1986 and the Ph.D. degree in computer science from the University of Karlsruhe, Karlsruhe, Germany, in 1992.

He is Assistant Professor of Computer and Information Science, University of Pennsylvania, Philadelphia, affiliated with the interdisciplinary GRASP laboratory. Prior to his current appointment he was with the Cognitive Systems Group, University of Kiel. Currently, his research centers on omnidirectional vision and vision techniques for tele-immersion and augmented reality.

Prof. Daniilidis was the recipient of the 2001 Motor Company Award for the Best Penn Engineering Faculty Advisor. He was the chair of the IEEE Workshop on Omnidirectional Vision 2000. He is the co-chair of the computer vision TC of the Robotics and Automation Society and has been reviewing for the main journals, conferences, and funding panels in computer vision.