blue-c: A Spatially Immersive Display and 3D Video Portal for Telepresence

Markus Gross Stephan Würmlin Martin Naef Edouard Lamboray
Christian Spagno Andreas Kunz Esther Koller-Meier Tomas Svoboda Luc Van Gool
Silke Lang Kai Strehlke Andrew Vande Moere Oliver Staadt*
ETH Zürich

Figure 1: Panoramic picture of the blue-c portal showing the projection screens in transparent mode, the outer scaffold with attached cameras, the LED arrays, and the LCD stereo projectors.

Abstract

We present *blue-c*, a new immersive projection and 3D video acquisition environment for virtual design and collaboration. It combines simultaneous acquisition of multiple live video streams with advanced 3D projection technology in a CAVE™-like environment, creating the impression of total immersion. The blue-c portal currently consists of three rectangular projection screens that are built from glass panels containing liquid crystal layers. These screens can be switched from a whitish opaque state (for projection) to a transparent state (for acquisition), which allows the video cameras to "look through" the walls. Our projection technology is based on active stereo using two LCD projectors per screen. The projectors are synchronously shuttered along with the screens, the stereo glasses, active illumination devices, and the acquisition hardware. From multiple video streams, we compute a 3D video representation of the user in real time. The resulting video inlays are integrated into a networked virtual environment. Our design is highly scalable, enabling blue-c to connect to portals with less sophisticated hardware.

Keywords: Spatially Immersive Displays, Virtual Environments, Graphics Hardware, 3D Video, Real-time Graphics.

*now at University of California, Davis.
Project webpage: http://blue-c.ethz.ch/

1 Introduction

1.1 Background and Motivation

Over the past decade, spatially immersive displays (SIDs) have become increasingly significant. SIDs enable users to work and interact with virtual spaces while being physically surrounded with a panorama of imagery. Among the variety of available SIDs, the most popular one is the Cave Automated Virtual Environment (CAVE™) [Cruz-Neira et al. 1993]. More recently, these SIDs have been extended by integrating multiple projection devices and screens into conference rooms or office spaces. Some of these SIDs have already been turned into products and have proved their usefulness in a wide spectrum of applications, including architecture, art, medicine, and the automotive industry. Although many alternative display technologies are currently available, multi-projector SIDs are clearly superior with regard to display quality, degree of immersion, resolution, field of view, and ergonomics.

Many applications, however, demand support for telecollaboration. Hence, in recent years, much research has been devoted to enhance SIDs by a variety of technical solutions to support telepresence in combined virtual and real environments. Such mixed reality systems are essentially proposed to augment immersion and realism. Central to all approaches is the seamless and realistic integration of remotely located users into the synthesized virtual space. Early approaches build on 2D video conferencing and utilize single cameras to acquire video streams of the collaborators. By reprojecting the video onto geometry, such systems create virtual meeting places for remotely located users. However, in addition to their inherently limited expressiveness, 2D video inlays impose severe constraints on the natural interaction of the user with the virtual world. The National Tele-Immersion Initiative [Sadagic et al. 2001] constitutes the most comprehensive program to address all these aspects.

The latest advances in real-time scene acquisition and 3D video have been exploited to create much more realistic 3D scans of actors or entire scenes. The seamless and robust combination of multiple projection devices with 3D scene acquisition, however,

poses great technical challenges, some of which have been set forth in UNC's Office of the Future [Raskar et al. 1998]. Here, considerable research has been pursued to build smart rooms with multiple cameras and projectors. Despite its practical importance, the 3D video-based enhancement of highly scalable SIDs such as the CAVE™ has yet to occur because of the daunting technical problems involved. In particular, the tough problem of simultaneous, bidirectional projection and acquisition has not been attacked yet.

blue-c presents a solution to this problem. Our work was motivated by the desire to combine the advantages of the total immersion experienced in a CAVE™ with simultaneous real-time 3D video acquisition and rendering from multiple cameras. By developing a novel combination of projection and acquisition hardware and by designing and refining a variety of algorithms, we create photorealistic 3D video inlays of the user in real time and at a quality that goes well beyond the known 2D video billboards or sprites.

The blue-c project is highly interdisciplinary in nature and requires expertise from computer graphics, vision, communication engineering, mechanical engineering, and physics. It took three years and the dedication of 20 researchers to bring this project to completion.

1.2 Contributions

In this paper we present the major milestone and technical achievement of the blue-c project: the extension of the blue-c hardware and its combination with a real-time 3D video pipeline for simultaneous acquisition and projection.The technical contributions can be summarized as follows:

- Spagno and Kunz [2003] designed a novel concept for projection technology that uses *active screens* with liquid crystal (LC) layers inside. By employing screens with switchable transparency and by synchronizing them with other hardware components, we enable the cameras to "look through" the walls of the system. We extended this initial concept by adding pulsed LED illumination to improve image quality and to facilitate the acquisition. We also optimized the camera layout, timings, and operating points critical for proper synchronization of the hardware components, including projectors, LC panels, shutter glasses, cameras, and active illumination.
- We devised a novel 3D video processing pipeline tailored to our needs. It utilizes 3D irregular point samples as video primitives and combines the simplicity of 2D video processing with the power of 3D video representations. This allows for highly effective encoding, transmission, and rendering of 3D video inlays.
- We designed a network communication architecture for the blue-c system. It offers various services for managing the nodes of the distributed and heterogeneous setup and implements communication channels for real-time streaming, suited for 3D video, audio, and synchronization data. The blue-c API presented in Naef et al. [2003] permits application programmers to exploit all blue-c-specific hard- and software features in a user-friendly way.

While blue-c is primarily intended for high-end collaborative SIDs and for telepresence, the setup is highly scalable, allowing users to adjust the number of cameras and projectors to the needs of the application.

2 Background and Related Work

In this section we will discuss prior work in display environments, 3D video acquisition methods, and combinations of both relevant to blue-c.

Display systems. Examples of room-sized multi-projector SIDs with up to six planar display walls include the CAVE™ [Cruz-Neira et al. 1993] and its numerous variants. Related systems comprise tiled display walls, such as the PowerWall or the WorkWall™. While those systems feature a large display area at high resolution, they do not physically surround the viewer and, thus, are not fully immersive. The Elumens VisionDome is a single-projector spherical SID for up to 45 viewers. A major disadvantage of single-projector SIDs, however, is their limited resolution. Other (semi-) immersive displays include the single projector responsive workbench [Krüger et al. 1995; Agrawala et al. 1997] and the Holobench™, an L-shaped projection table with two orthogonal projection surfaces. An alternative to projection-based systems are semi-immersive autostereoscopic displays such as Perlin et al. [2000].

Acquisition and reconstruction. In the related literature we see many methods for the processing of multi-camera video images to 3D videos. As opposed to post-processing approaches like Moezzi et al. [1996] or Würmlin et al. [2002], interactive telepresence demands real time. For instance, infrared-based structured light [Davis and Bobick 1998] was successfully used for such applications. Narayanan et al. [1998] employ a triangular texture-mapped mesh representation. Pollard and Hayes [1998] utilize depth map representations to render real scenes from new viewpoints by morphing live video streams. Shape-from-silhouette techniques can be used to reconstruct visual hulls [Laurentini 1994] in real time. Two fast methods are the image-based visual hull (IBVH) introduced by Matusik et al. [2000], which takes advantage of epipolar geometry to build a layered depth-image representation, and the polyhedral visual hull [Matusik et al. 2001], which constructs a triangular surface representation.

Combinations. While there are a considerable number of projection displays and 3D video acquisition methods, relatively little work has been devoted to integrating both into a single system. The TELEPORT environment [Gibbs et al. 1999] is an example of a telepresence system with user integration by means of 2D video avatars. The system uses delta-keying to separate the remote participant from the background. 3D video avatars are used by the National Tele-Immersion Initiative [Sadagic et al. 2001], where the remote participant representations are derived from trinocular stereo [Mulligan and Daniilidis 2000]. Subramanian et al. [2002] developed a video avatar system for networked virtual environments, which maps a view-dependent video texture on a preacquired head model using head tracking information. In the Office of the Future [Raskar et al. 1998], depth information is obtained by a structured light technique. In both systems, the surrounding background environment is scanned off-line using a laser range finder and rendered as a static scene. A similar concept has been presented by Kauff and Schreer [2002] that proposes an immersive 3D video conferencing system with real-time reconstruction.

3 Concept and Systems Setup

The conceptual components of our architecture are presented in Figure 2. Three twin LCD projectors with additional LC shutters are utilized to generate a CAVE™-like immersive display empowered with active stereo. One of the technical core novelties of blue-c is the use of an *actively shuttered projection screen* allowing the video cameras to "see through the wall" during frame acquisition (Section 4.2). As we will discuss, this solves a variety of problems regarding 3D reconstruction and projection. Additional active LED illumination (Section 4.4) produces calibrated lighting conditions during the acquisition phase, which greatly improves the texture and color quality of the 3D video. Video frame capturing is triggered at a fixed rate in a small time slot between left and right eye projection. During acquisition, the shutter glasses are switched to opaque for both eyes, as are the projector's LC shutters (Section 4.5). These components demand precise synchonization by special-purpose microcontroller hardware (Sec-

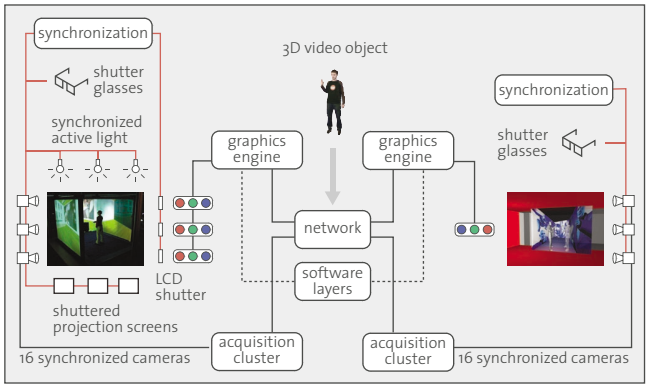

Figure 2: Conceptual components of the blue-c architecture including hardware and software. We also display the lines (red) which synchronize the hardware components as well as the connection to a remote site.

tion 4.3). For image generation, we currently use an SGI Onyx 3200. Additional hard- and software components accomplish head tracking, spatial audio, voice communication, and 3D user interaction.

We designed a novel 3D video engine to process multiple video streams on a Linux PC cluster (Section 5.1). After background substraction and silhouette extraction (Section 5.3), we compute a so-called 3D video fragment representation of the user in real time (Section 5.4). This format progressively encodes the 3D stream and facilitates efficient rendering and 3D compositing. Two or more portals can be connected by a high speed IP network. The blue-c communication layer (Section 5.5) handles the transmission of 3D video streams, along with the events from the collaborative application. A high level, distributed scene graph API gives users full access and control over all blue-c-specific features (Section 5.6). Although blue-c constitutes a high-end SID, our design allows connections to remote portals with much less technical sophistication.

The remainder of the paper is devoted to a discussion of the hardware (Section 4) and software components (Section 5) of blue-c.

4 Hardware Components

In this section, we present a high level overview of the blue-c hardware. We will emphasize the technical modifications and extensions of the initial concept of Spagno and Kunz [2003] including timings, operating points, active lighting, and camera layout.

4.1 Configuration and Camera Arrangement

We built the blue-c portal as a three-sided stereo backprojection system, providing a well balanced trade-off between constructive complexity and degree of immersion. Active floor and ceiling are omitted but could be added to future versions. We chose a planar, cube-like configuration because the glass screens (Section 4.2) have not been available as curved surfaces. We cannot employ passive stereo based on polarized glasses, because it requires the projection screens to retain the polarization. Futhermore, we take advantage of the modified shutter glasses to protect the user's eye from the illumination (Section 4.3). As a central part of our design, we place most of the cameras *outside* the SID space, as illustrated in Figure 3a. Besides the issues discussed above, this strategy leaves us much more freedom to optimize camera positions with regard to 3D reconstruction. The current setup features 16 VGA-resolution Dragonfly™ Firewire cameras from Point Grey Research, 11 of which acquire video streams from outside for 3D reconstruction. The five remaining cameras are attached to the four upper corners of the screen and to the ceiling. While having less

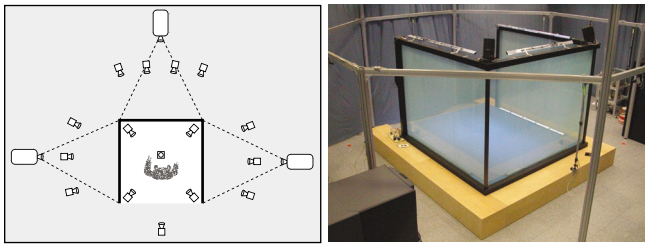

Figure 3: Picture illustrating the projector-camera arrangement of blue-c. a) Schematic view. b) Real system with scaffold to attach the cameras.

optimal positions for shape reconstruction, they greatly facilitate texture acquisition and calibration (Section 5.2). Based on experimental evidence and software simulation, we found that this arrangement achieves optimal quality both in geometric and photometric reconstruction. In Figure 3b we show our current setup including the scaffold to which cameras can be attached at arbitrary locations. Blue fabric curtains in the background further support background subtraction (Section 5.3).

4.2 Active Panels

In the above setup it is easy to see that the projection screens occlude the user from the outside cameras. Spagno and Kunz [2003] solve this problem by using IsoLite™ glass projection screens, which have a *phase dispersed liquid crystal* (PDLC) layer inside [Lampert 1999]. This layer can be electrically switched from an opaque state to a transparent state. The glass panels have an additional anti-reflective coating to attenuate highlights and reflections. By shuttering the screen, we accomplish time multiplexing between image generation and image aquisition. In our current implementation, each projection wall is composed of three 950 mm × 2,240 mm LC-panels from ISOCLIMA S.P.A., weighing 80 kg and costing $3,300 US per panel. The 2 mm gap between the panels is permanently opaque and hence does not significantly deteriorate the display quality.

These panels are actually not constructed for fast periodic switching. Each LC layer constitutes a capacity of 13 µF, which has to be charged and discharged in each cycle. The internal electric field forces the molecules of the LC panel to arrange so that the display becomes transparent. The speed of the transition from opaque to transparent can be actively contolled by applying a voltage of +/– 60 V to the panel's electrodes and by limiting the currents. The rearrangement of the molecules from transparent to opaque, however, is accomplished by internal relaxation (0 V). This property of the LC layer cannot be influenced actively. When applying a periodic trigger signal, we found that the transition from opaque to transparent occurs in about 3 ms, whereas the inverse transition is much slower due to relaxation. The polarity of the voltage signal has to be toggled in each cycle to avoid DC components that would damage the LC layer.

The degree to which we charge and discharge the capacitors along with the relaxation determines the residual opacity and transparency in the projection and acquisition stages. As a result, the frequency response of the panels exhibits a lowpass behavior that imposes a limit onto the shutter frequencies. We conducted various experiments with small and large panel sizes of different manufacturers and found that the large size LC layers still work fine for 62.5 Hz (16 ms period) and the given duty cycle of 1 : 3.44. Figure 6a depicts the frequency response curve of the opacity-transparency amplitude of one of the panels, measured in relative units from 0 to 1. The screen still achieves close to 70% opacity at the operating point, which turns out to be sufficient for high quality projection and reconstruction. Our experiments have also

shown that the fields generated by the switch signals do not interfere with any other electromagnetic component of blue-c, such as audio or tracking.

4.3 Synchronization and Timing

We acquire the video frames in a small time slot between the projection frames for the left and right eye. Figure 4 gives the timing diagram of the resulting acquisition and projection cycles. As depicted, we "open" the walls during a short time slot of typically 3.6 ms to acquire the video frame. During the acquisition time slot the shutters for both the user's eyes and for the projectors are entirely opaque to avoid disturbing artifacts, in particular highlights and flickering. Thus, we overlay the acquisition phase without having to modify the projection cycles. Currently, processing speed limits the video acquisition rate to 8.9 Hz. Hence, the system grabs frames in every 7th window. Note that the trigger signals must be shifted to account for the individual delays of the components. A custom synchronization device was designed which generates the required timing and trigger patterns for all components of blue-c.

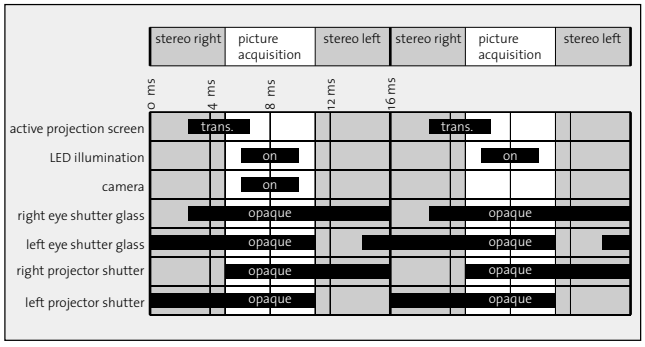

Figure 4: Timing diagram including the trigger signals for all actively synchronized hardware components. Note the shifted trigger signals which account for switching delays of the panels.

4.4 Active Illumination

Another great technical challenge when combining acquisition and projection is the conflicting requirements concerning lighting conditions. In a dark environment, a user will experience a high degree of immersion, whereas a bright object allows for proper background separation, texture acquisition, and 3D reconstruction (see also Section 5.3). To demonstrate this aspect we compare the interior of blue-c with and without active illumination in Figure 5.

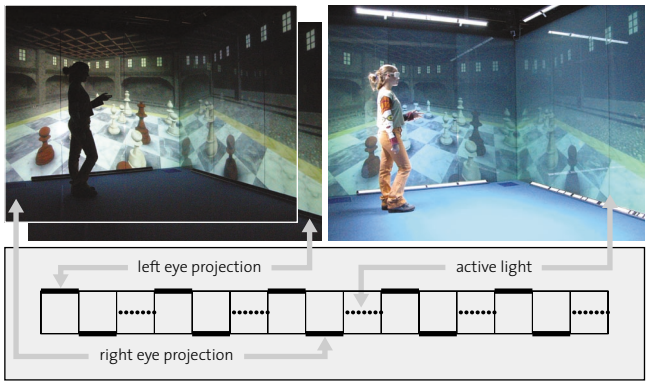

Figure 5: Lighting conditions in the blue-c portal with and without active illumination. The timing diagram illustrates the projection phases and light pulses.

In principle, the light scattered from the outside could be used to illuminate the user. Practice has shown, however, that conventional illumination is either not bright enough to guarantee proper appearance samples or not weak enough to guarantee high quality projection. We greatly improve the quality of the video streams by actively illuminating the platform using pulsed calibrated light sources. More sensitive cameras are not seen as an alternative. Besides their high costs, most of them are monochromatic and do not allow for color images.

The lighting system of blue-c is synchronized with the video frame acquisition and emits a light flash in each time slot at 62.5 Hz, as we illustrate in the timing diagram of Figure 4. The flashes are well above the fusion frequency of the human eye. We first investigated stroboscopic sources. Due to their very short flashes of about 20 µs, the total amount of luminous energy, as integrated by the CCD camera sensor, is relatively low. Since strobes do not allow control of the length of the flash, a very high intensity pulse was needed.

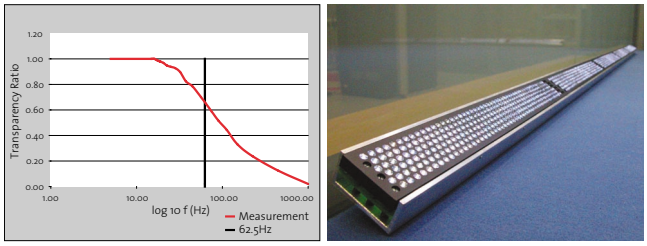

Figure 6: a) Frequency response of the relative opacity of the PDLC panels. b) LED array module used for active illumination.

The high voltage discharge, however, caused unacceptable electromagnetic noise that was difficult to cope with. We thus designed and manufactured light sources based on white LED arrays, as shown in Figure 6b, which are attached to the top and bottom frames of the portal. The LEDs have very short switching times of 0.5 µs, have long life cycles, and do not cause noticable electromagnetic interference. The blue-c portal accomodates a total of 10,000 LEDs grouped into 32 clusters, with 2,200 cd/LED at an aperture of 50 degrees. To avoid deteriorating reflections on the projection screens we adjusted the arrays to illuminate the user only.

4.5 Stereo Projection and Construction

Shutter Glasses. As we show in Figure 4, the triggering of shutter glasses has to be modified to produce an opaque phase during the acquisition time slot. This is necessary to protect the user's eye from the active light pulses. We modified the firmware of NuVision 60GX active shutter glasses to allow for the overlay phase. For the user's convenience, the shutters are triggered using a wireless IR emitter that transmits the trigger signal.

Projectors. Active stereo projection is generated by three pairs of Sanyo XF12 LCD projectors with 3,500 ANSI lumen each, which are mounted on racks. They are placed at a distance of 3.22 m from the screens. Custom-made external ferro-electric LC shutters (100 µs switching time) with an aperture size of 140 mm × 140 mm are mounted in front of the 1:1.2 lenses, allowing us to alternate the projections as required for active stereo. By applying the same trigger signal to both the shutter glasses and to the projection systems' LC shutters, we attenuate the projection beams during the acquisition stage to a very large extent. This further avoids interferences between projection and acquisition. As an alternative, we could use commerical active stereo DLP products, but these were not available at the time we started this project. In addition, the sequential color scheme of the DLPs makes synchronization much more involved and less flexible.

Construction. The mechanical construction is custom designed and hosts most hardware components including the screens, the tracking system, loudspeakers, active illumination, and IR-emitters [Spagno and Kunz 2003]. Due to the weight of the screens (720 kg), we designed and built a beech platform with a total height of 540 mm. The frames for the panels are made from fiber-reinforced composite to guarantee sufficient stiffness to hold the panels safely in place. The outer camera scaffold is made of $40\text{ mm} \times 80\text{ mm}$ and $110\text{ mm} \times 110\text{ mm}$ aluminum beams. The chosen materials ensure accurate functioning of all electrical and magnetic components. Integrated cable channels hide most of the wiring from the user. A panoramic view of the blue-c portal is given in Figure 1.

4.6 Tracking, I/O, and Spatial Audio

The blue-c portal utilizes a conventional Ascension Flock of Birds six degrees-of-freedom magnetic tracking system. The extended range transmitter is integrated into the wooden floor of the blue-c structure. Three sensors serve for tracking the user's head, a 3D mouse, and a Wand 3D mouse with a joystick. We decided to use a standard tracking system to minimize development time and to interface to a wide range of input devices. For software development and testing, mouse and keyboard input is also supported. Given our glass, carbon, and wood construction, we never encountered interference with the tracking system in our experiments. As an alternative, one might use some of the cameras for vision-based tracking, which is subject to future work. Spatialized sound rendering, including room simulation, is provided by a sound server that runs either on the graphics server or on an independent PC. This concept is presented in Naef et al. [2002]. For sound output, blue-c hosts standard studio hardware and drives six active Yamaha MSP5 loudspeakers and a Yamaha SW10 subwoofer. We use standard Sennheiser wireless microphones for speech recording and transmission.

5 Software Components

In this section we will discuss the blue-c software modules, emphasizing the 3D video pipeline.

5.1 Real-Time 3D Video Processing

After analyzing different concepts for real-time 3D reconstruction (see Section 2) we found that silhouette-based methods best serve our purposes. They offer an excellent trade-off between reconstruction quality and efficiency. Therefore, we employ a point-based variant of the image-based visual hull [Matusik et al. 2000]. The conceptual components and individual processing stages of our 3D acquisition and video pipeline are presented in Figure 7.

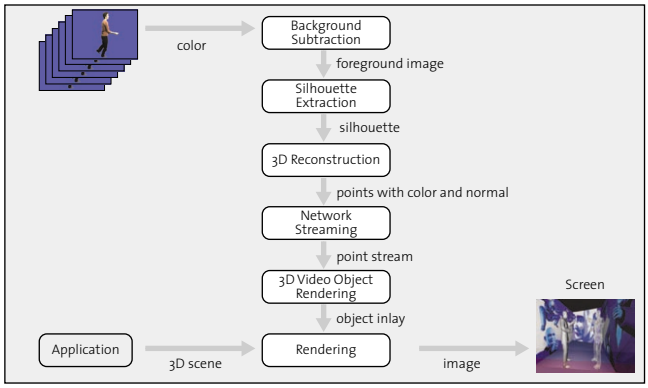

Figure 7: Conceptual components of the 3D reconstruction and video processing pipeline.

In a preprocessing step, before the actual session starts, the system has to be calibrated. This includes the computation of extrinsic and intrinsic camera parameters, the radial lens distortion, and a color calibration (Section 5.2). During run-time, input video images are processed in parallel on a PC cluster. Foreground pixels are extracted by background substraction and a multi-scale silhouette of the user is computed (Section 5.3). These silhouettes serve as an input to extract a 3D point-based video representation (Section 5.4) from visual hulls. Our concept of *video fragments* exploits the spatio-temporal coherence of the video streams using interframe prediction and a differential update scheme [Würmlin et al. 2003]. This 3D video stream is progressively transmitted over the network, calling services from the blue-c communication layer (Section 5.5). The simplicity of our 3D video format permits fast and high quality display using point-based 3D rendering technology [Zwicker et al. 2001]. Compositing with triangle-based application geometry can be easily accomplished using the z-buffer. We defer shading operations to accelerate rendering. Additional 3D video recording technology allows us to record, store, and postprocess 3D video streams as needed. The performance of the 3D video pipeline is discussed in Section 6.

5.2 Calibration and Correction

Camera Parameters. In order to create a Euclidean 3D model, all cameras have to be fully calibrated with respect to a single world coordinate system. We use an automatic procedure based on self-calibration [Pollefeys et al. 1999]. A person waves a standard laser pointer in the darkened blue-c, filling the working volume with virtual points. Geometric constraints are used to clean the set of points in the different views. False points are removed and missing points are added. Their projective structure is computed by rank-factorization and refined through bundle adjustment. This self-calibration procedure yields all parameters of the pinhole camera model. The reprojection error is mostly below one pixel. Despite our outside camera placement, we still have relatively short distances from the user to the cameras (2.5 m - 3 m). Hence, we have to correct for radial lens distortion as well. To this end we employ the Caltech camera calibration toolbox for estimation and Intel's Open Computer Vision Library for correction.

Color Correction. blue-c requires correcting the colors of different cameras. Some cameras see through the projection screens, some do not. Different viewing directions with respect to the screens lead to slightly different degrees of transparency and dispersion. Our approach uses color distributions from the unobstructed corner and ceiling cameras to adjust the colors of the outside cameras accordingly. The color transformations are carried out in the l,α,β space, as proposed by Reinhard et al. [2001]. The correction proceeds in two steps. First, the color space of the source image is decorrelated and normalized. Second, the means and variances of the target images are used for inverse conversion into RGB space. This transformation is actually non-linear and is linearized for fast processing. In our experiments we achieved non-perceivable approximation errors with maxima in very dark or very saturated areas where fine differences in color are less important. The color transformation is different for each camera and can be precomputed off-line. Only the pixels that are used for 3D reconstruction are transformed on-line using a 3×3 matrix. Figures 8b and 8c compare uncorrected and corrected frames.

5.3 Background Subtraction and Silhouette Extraction

Background Processing. A major selection criterion for the background subtraction algorithm was the real-time constraint. After various experiments, we decided to adapt an algorithm whose details can be found in Mester et al. [2001]. Our method

assumes a static background and detects changes using a statistical collinearity criterion. The color values within a small neighborhood of each pixel are rearranged into a vector that is compared to the average values from the background. The collinearity between the two vectors is measured in an optimal statistical way. If it is smaller than a threshold, which is learned from an analysis of the background images, the pixel is considered as foreground. A blue anti-reflective fabric curtain helps us improve the performance of the method. The image sequence in Figure 8 presents a typical video frame as acquired from one of the blue-c cameras with a coarse scale silhouette before (a), and after background subtraction (b). A potential alternative is infrared-based silhouette extraction, as proposed in Debevec et al. [2002]. In this case, however, the glass screens would force us to place all emitters and cameras inside the stage, which contradicts our approach.

Adaptive Silhouettes. As the visual hull computation time depends on both the number of contours and contour samples, we devised a multi-scale contour extraction algorithm that balances speed and quality. The method progressively marches the pixels of the silhouette boundaries and stores edge endpoints. The contour is adaptively extracted from the intersections of multi-scale cells covering the segmented image. A user-controlled parameter α determines the scale and thus the accuracy of the contour. In practice, we set $\alpha = 1$ or 2 and obtain 2×2 or 4×4 grids. This trades 3D video quality for performance. A typical example for an adaptively refined silhouette is given in Figure 8b and 8c.

Figure 8: a) Input video frame. b) Background subtraction and coarse silhouette. c) Adaptively refined silhouette and color correction.

5.4 3D Reconstruction and Video Representation

Conceptually, our 3D video representation generalizes 2D video pixels towards 3D irregular point samples, which combines the simplicity of conventional 2D video processing with the power of more complex 3D representations. A *video fragment* is a point sample of the surface of the 3D object with additional geometric and appearance attributes, such as surface normal and color.

In order to exploit the spatio-temporal coherence of consecutive 2D video frames from multiple cameras, we propose a differential update scheme for the point representation. For each point update we transmit a specific operator such as *insert*, *update*, or *delete*, which accounts for the different types of input changes. Interframe prediction is carried out in the reference images to avoid computationally expensive 3D processing, as in other approaches to 3D video [Vedula et al. 2002]. *Inserts* and *deletes* are classified according to contour changes. Potential candidates for an *update* are all pixels that have been inserted in previous frames and that are still valid. To optimize processing, we divide them into three categories: *unchanged*, *color change*, and *geometry change.* A simple color differencing is used for detection of color changes. Changes of the geometry are analyzed on spatially coherent clusters of pixels in image space. A small number of randomly sampled pixels is used to determine whether 3D information for the entire pixel cluster must be recomputed. The described operators are merged into a 3D video stream and transmitted to the remote side where a flat data representation is dynamically updated. This datastructure is displayed using point-based rendering. Figure 9a shows a classified video frame.

To further accelerate the 3D video processing, the actual 3D reconstruction for a desired view is carried out using a subset of the reference views. For each frame we dynamically select a set of *active cameras* as reference views based on their pose deviation from the desired view. Furthermore, we define an activity level for each active camera serving as a basis for a smooth blending. This is accomplished by weighted averaging, similar in spirit to Unstructured Lumigraph Rendering [Buehler et al. 2001]. The aforementioned operators are computed only on the active cameras, making our representation intrinsically view-dependent. Adding more active cameras to the reconstruction set enables us to smoothly shift between a view-dependent and a view-independent representation. In our implementation we utilize three active cameras. We refer the reader to Würmlin et al. [2003] for an in-depth technical discussion of the 3D video engine.

Each of the 16 cameras is connected to a node of a PC cluster that runs the image-based algorithms in parallel. An additional node performs the visual hull reconstruction and 3D streaming. The 3D representation resides at the remote side and is concurrently accessed by the 3D video stream and by the local renderer.

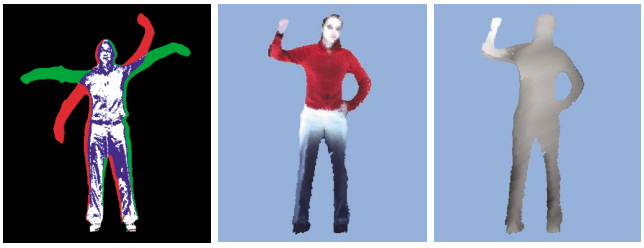

Figure 9: A 3D video inlay computed by our method. a) Pixel classification. Red indicates *delete*, green indicates *insert*, blue indicates *update color*, white indicates *update position* candidates. b) Point-rendered view of a 3D video inlay. c) Corresponding depth map.

5.5 Networking and Communication

blue-c is a complex distributed environment, composed of many heterogeneous subsystems. It requires an efficient communication framework to handle not only the data generated by our 3D video engine, but also all other data streams common to most networked and collaborative virtual environments. These include *bulk data*, i.e., the initial scene description and application data; *event and message-based communication*, i.e., user-triggered modifications of the shared virtual scene; *real-time audio* for voice communication; and *tracking and system information* that is exploited for resource optimization and rendering.

We designed the blue-c communication and system layer for this purpose; the architecture is displayed in Figure 10. The data transmission within one blue-c portal, as well as in between portals, is achieved with the advanced programming model of the CORBA architecture and its Audio/Video Streaming Service. Control information and non-real-time data is transmitted using traditional CORBA remote method invocations. The real-time data, however, is transmitted via out-of-band streams, using transfer protocols such as UDP or RTP. Hence, no penalizing overhead is introduced for performance-critical data. The blue-c core communication and system software is developed using the TAO/ACE framework [Schmidt].The nodes of a single blue-c portal run on a standard 100 Mbps or 1000 Mbps Ethernet network. Remote portals are

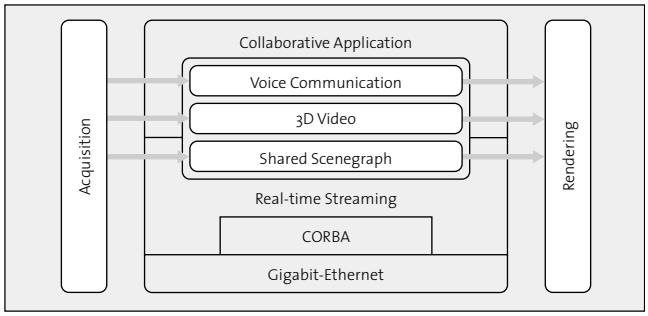

Figure 10: Architecture of the blue-c communication and system layer.

typically interconnected via Gigabit Ethernet links, but ATM OC3 is also supported. The system allows for full-duplex 3D video, voice communication, and collaborative user interaction on the distributed shared scene graph.

5.6 Application Programming Interface

The blue-c API provides an application development environment that offers flexible access to all blue-c features, including graphics and sound rendering, device input, 3D video, and scene distribution. These subsystems are provided as services and managed by the blue-c API *core* class. The API is designed to support an arbitrary number of portals with different hard- and software configurations. A conceptual overview of the API is given in Figure 11.

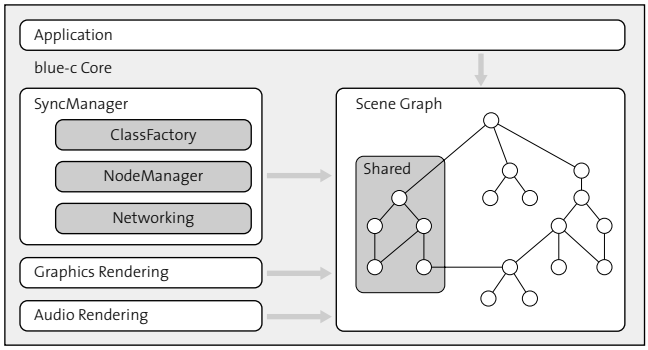

Figure 11: Overview of the blue-c API.

The core class representing the complete virtual world is a distributed, shared scene graph based on SGI OpenGL Performer [Rohlf and Helman 1994]. We chose Performer as a basis platform because it natively supports multi-pipe, multi-processed rendering with very high throughput.

The scene graph has been enhanced with special purpose nodes, which allow us to treat 3D video objects as if they were regular geometry objects. They support transformations, culling, and correct occlusions. The 3D video fragments are rendered in a callback using optimized OpenGL code. Audio nodes are also part of the scene and permit attaching fully spatialized sounds to objects. Active nodes support object interaction and the triggering of animations. Conventional 2D video stream rendering for both local files and remote cameras is implemented as customized texture objects. All nodes of the scene include an interface to accomplish the replication and synchronization of the scene across all participating sites. We achieve immediate local visual response to object manipulations while still keeping the scene consistent utilizing a relaxed locking scheme.

The blue-c core takes full advantage of the parallel-processing hardware of the graphics server. The decoding and preprocessing of 3D video fragment streams is performed in a seperate process that runs in parallel to the rendering processes. Also, video decoding, audio rendering, occlusion culling, and tracker device handling each have their own processes. The individual processes communicate by means of a message passing system, making distribution across a cluster of PCs feasible. A detailed technical discussion of the blue-c API is provided in Naef et al. [2003].

6 Performance and Applications

Our 3D video processing pipeline currently reconstructs video inlays composed of approximately 25k points at 5fps or 15k points at 9 fps. The 3D video representation is updated by a stream of 30k to 200k 3D point operations per second, depending on the degree of inter-frame coherence. In our performance analysis we measured that, on the average, about 25% of all fragments are updated per frame. The measured distribution according to the individual operators amounts to 25% inserts, 50% updates, and 25% deletes. The required bandwidth for the 3D video stream varies between 2.5 and 12.5 megabits per second at the given resolution. The overall system latency from 2D acquisition to remote 3D rendering lies between 3 and 5 frames. It was not fully optimized by the time we finished this paper.

While it is not possible to capture the visual quality of the projection perceived by the user, the following still images and the accompanying videotape give a good feeling of the overall system performance. Note specifically that the projector highlights apparent in the video, and in some of the pictures, are attenuated to a very large extent for the user's eye. To better demonstrate the display quality, we used a monomode projection for the screen shots. All of the examples and applications shown in this paper were created using the blue-c API.

To demonstrate the core functionality of blue-c we implemented a "3D mirror" hall. Here, the user experiences herself in a virtual world in real-time. She can freely navigate and see herself in full 3D from arbitrary viewing directions. Figure 12 gives a view into the blue-c portal illustrating the immersive projection and the user interaction. The corresponding view being projected onto the LC panels is displayed in Figure 16. The application context of this example is fashion, where a remote model presents the latest fashion design in a virtual shop. In this application, the 3D representation constitutes an important component for interpersonal communication and for conveying the intention of the designer. Similarly, Figure 13 and Figure 17 illustrate an art performance featuring a dancer in a virtual particle universe. Another sales application is depicted in Figure 14 and Figure 18, in which a car sales person is presenting a luxury car. The user is immersed in the virtual scene while her 3D video is composited and transmitted to the remote projection site. To demonstrate the quality of the 3D video, we reprojected the final scene onto the LC panels. Finally, Figure 19 presents the resolution limits of the 3D video inlay by displaying an extreme close-up. We rendered the circular primitives opaque to illustrate the sampling distribution. The corresponding close-up of the real user's body is depicted in Figure 15. The color distribution of the splat primitives provides a coarse sampling of the high-resolution texture pattern of the skirt. Note that the colors of the images could not be adjusted for printing.

Besides camera resolution, the quality of the 3D video depends on various factors. First, the 3D visual hull computations are currently carried out on two processors in parallel, posing a limit onto the resolution of the point stream. The addition of more compute nodes to the reconstruction will further improve the performance. Second, the LED arrays have to be adjusted and dimmed thoroughly to avoid overillumination of parts of the user's body. Third, while the segmentation works fine, mismatches occassionally produce spurious sillhouettes that degrade the accuracy of reconstruction in some frames. Here, a heuristic estimation of the quality of the silhouettes could be used to discard corrupted frames from

individual cameras. In all the examples from above, we actually transmitted the 3D video over the network to a remote client for rendering and projection.

7 Conclusions and Future Work

We have presented blue-c, a novel hard- and software system that combines the advantages of a CAVE™-like projection environment with simultaneous 3D video capture and processing capabilities. The system is primarily designed for high-end remote collaboration and presentation. To build the system, we devised a variety of novel concepts, both in hardware and in software. Our architecture is highly scalable, allowing the user to adapt the number of cameras or projectors as needed. Given the price of modern PC clusters, cameras, and projection hardware, blue-c is not much more expensive than a conventional CAVE™. As a major limitation, our system is currently confined to a single user per portal. An extension to multiple users clearly is a high priority for future research. We plan to investigate different concepts for real-time 3D reconstruction. Furthermore, our multi-camera arrangement is well suited for vision-based tracking, and we will experiment with next generation projection technology. Finally, we are exploring new metaphors for human-computer interaction emerging from blue-c.

Acknowledgements

The blue-c project was funded by ETH grant No. 0-23803-00 as an internal poly-project. We would like to thank Maia Engeli, Ludger Hovestadt, Markus Meier, and Gerhard Schmitt for their continuing support and for many helpful discussions; Kuk Hwan Mieusset; our students Stefan Hösli, Nicholas Kern, and Christoph Niederberger; and Jennifer Roderick Pfister for proofreading the paper.

References

AGRAWALA, M., BEERS, A. C., FRÖHLICH, B., HANRAHAN, P. M., MCDOWALL, I., AND BOLAS, M. 1997. The two-user responsive workbench: Support for collaboration through independent views of a shared space. In *Proceedings of SIGGRAPH 97*, pages 327–332, ACM SIGGRAPH / Addison Wesley.

BUEHLER, C., BOSSE, M., MCMILLAN, L., GORTLER, S., AND COHEN, M. 2001. Unstructured lumigraph rendering. In *Proceedings of SIGGRAPH 2001*, pages 425–432. ACM Press / ACM SIGGRAPH / New York.

CRUZ-NEIRA, C., SANDIN, D. J., AND DEFANTI, T. A. 1993. Surround-screen projection-based virtual reality: The design and implementation of the cave. In *Proceedings of SIGGRAPH 93*, pages 135–142, ACM SIGGRAPH / Addison Wesley.

DAVIS, J. W. AND BOBICK, A. F. 1998. Sideshow: A silhouette-based interactive dual-screen environment. Technical Report 457, MIT Media Lab.

DEBEVEC, P., WENGER, A., TCHOU, C., GARDNER, A., WAESE, J., AND HAWKINS, T. 2002. A lighting reproduction approach to live-action compositing. In Proceedings of SIGGRAPH 2002, pages 547–556. ACM Press / ACM SIGGRAPH / New York.

GIBBS, S. J., ARAPIS, C., AND BREITENEDER, C. J. 1999. TELEPORT - towards immersive copresence. *Multimedia Systems*, 7(3):214–221. Springer.

KAUFF, P. AND SCHREER, O. 2002. An immersive 3D video-conferencing system using shared virtual team user environments. In *Proceedings of CVE'02*, pages 105–112. ACM Press.

KRÜGER, W., BOHN, C.-A., FRÖHLICH, B., SCHÜTH, H., STRAUSS, W., AND WESCHE, G. 1995. The responsive workbench: A virtual work environment. *IEEE Computer*, pages 42–48. IEEE Computer Society Press.

LAMPERT, C. 1999. The world of large-area glazing and displays. In *Switchable Materials and Flat Panel Displays*, pages 2–11. SPIE.

LAURENTINI, A. 1994. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162. IEEE Computer Society Press.

MATUSIK, W., BUEHLER, C., AND MCMILLAN, L. 2001. Polyhedral visual hulls for real-time rendering. In *Proceedings of Twelfth Eurographics Workshop on Rendering*, pages 115–125. Springer.

MATUSIK, W., BUEHLER, C., RASKAR, R., GORTLER, S. J., AND MCMILLAN, L. 2000. Image-based visual hulls. In *Proceedings of SIGGRAPH 2000*, pages 369–374. ACM Press / ACM SIGGRAPH / New York.

MESTER, R., AACH, T., AND DÜMBGEN, L. 2001. Illumination-invariant change detection using statistical colinearity criterion. In *DAGM2001*, number 2191 in LNCS, pages 170–177. Springer.

MOEZZI, S., KATKERE, A., KURAMURA, D. Y., AND JAIN, R. 1996. Immersive Video. In *Proceedings of the 1996 Virtual Reality Annual International Symposium*, pages 17–24. IEEE Computer Society Press.

MULLIGAN, J. AND DANIILIDIS, K. 2000. View-independent scene acquisition for tele-presence. In *Proceedings of the International Symposium on Augmented Reality*, pages 105–110. IEEE Computer Society Press.

NAEF, M., LAMBORAY, E., STAADT, O., AND GROSS, M. 2003. The blue-c distributed scene graph. In *Proceedings of the IPT/EGVE Workshop 2003*. ACM Press. To appear.

NAEF, M., STAADT, O., AND GROSS, M. 2002. Spatialized audio rendering for immersive virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology 2002*, pages 65–72. ACM Press.

NARAYANAN, P. J., RANDER, P., AND KANADE, T. 1998. Constructing virtual worlds using dense stereo. In *Proceedings of the International Conference on Computer Vision ICCV 98*, pages 3–10. IEEE Computer Society Press.

PERLIN, K., PAXIA, S., AND KOLLIN, J. S. 2000. An autostereoscopic display. In Proceedings of ACM SIGGRAPH 2000, pages 319–326. ACM Press / ACM SIGGRAPH / Addison Wesley.

POLLARD, S. AND HAYES, S. 1998. View synthesis by edge transfer with application to the generation of immersive video objects. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 91–98. ACM Press / ACM SIGGRAPH, New York.

POLLEFEYS, M., KOCH, R., AND VAN GOOL, L. 1999. Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1):7–25.

RASKAR, R., WELCH, G., CUTTS, M., LAKE, A., STESIN, L., AND FUCHS, H. 1998. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of SIGGRAPH 98*, pages 179–188. ACM SIGGRAPH / Addison Wesley.

REINHARD, E., ASHIKHMIN, M., GOOCH, B., AND SHIRLEY, P. 2001. Color transfer between images. *IEEE Computer Graphics and Applications*, 21(4):34–41. IEEE Computer Society Press.

ROHLF, J. AND HELMAN, J. 1994. IRIS Performer: A high performance multiprocessing toolkit for real-time 3d graphics. In *Proceedings of SIGGRAPH 94*, pages 381–395. ACM SIGGRAPH / Addison Wesley.

SADAGIC, A., TOWLES, H., LANIER, J., FUCHS, H., VAN DAM, A., DANIILIDIS, K., MULLIGAN, J., HOLDEN, L., AND ZELEZNIK, B. 2001. National tele-immersion initiative: Towards compelling tele-immersive collaborative environments. Presentation given at Medicine meets Virtual Reality 2001 Conference.

SCHMIDT, D.. Real-time CORBA with TAO (The ACE ORB). http://www.cs.wustl.edu/~schmidt/TAO.html.

SPAGNO, C. AND KUNZ, A. 2003. Construction of a three-sided immersive telecollaboration system. In *Proceedings of the IEEE Virtual Reality Conference 2003 (VR 2003)*, pages 37–44. IEEE Computer Society Press.

SUBRAMANIAN, S., RAJAN, V., KEENAN, D., SANDIN, D., DEFANTI, T., AND JOHNSON, A. 2002. A realistic video avatar system for networked virtual environments. In *Proceedings of Seventh Annual Immersive Projection Technology Symposium*.

VEDULA, S., BAKER, S., AND KANADE, T. 2002. Spatio-temporal view interpolation. In *Proceedings of the 13th ACM Eurographics Workshop on Rendering*, pages 65–75. ACM Press.

WÜRMLIN, S., LAMBORAY, E., STAADT, O. G., AND GROSS, M. H. 2002. 3D video recorder. In *Proceedings of Pacific Graphics 2002*, pages 325–334. IEEE Press.

WÜRMLIN, S., LAMBORAY, E., AND GROSS, M. H. 2003. 3D video fragments: Dynamic point samples for real-time free-viewpoint video. Technical Report No. 397, Institute of Scientific Computing, ETH Zurich.

ZWICKER, M., PFISTER, H., VANBAAR, J., AND GROSS, M. 2001. Surface splatting. In *SIGGRAPH 2001 Conference Proceedings*, pages 371–378. ACM Press / ACM SIGGRAPH / New York.

Figure 12: 3D mirror seen from a view into blue-c.

Figure 13: Art performance with real-time 3D visual feedback.

Figure 14: Car sales application with user and 3D video inlay.

Figure 15: Close-up of a part of the real user's body.

Figure 16: Snapshot of the 3D mirror hall application.

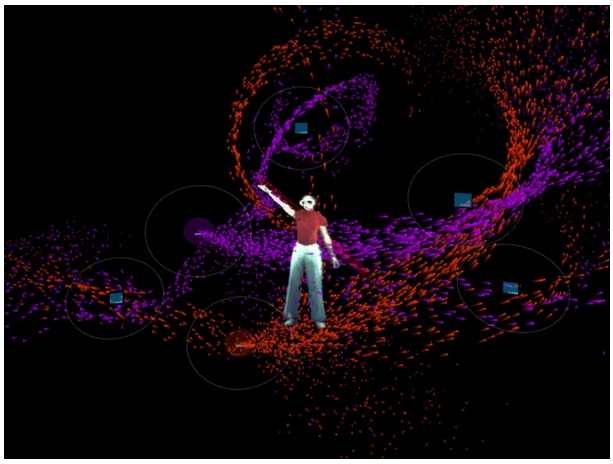

Figure 17: Snapshot of the the art performance application.

Figure 18: Snapshot of the car sales application.

Figure 19: Corresponding close-up of the 3D video inlay.