

blue-c

Räumlich-immersive Projektions- und 3D Video-Portale für Telepräsenz

Teil 2: Kommunikation, Systemleistung und Anwendungen

Markus Groß · Stephan Würmlin
Martin Naef · Edouard Lamboray
Christian Spagno

Kommunikation und Netzwerk

blue-c ist ein komplexes, verteiltes System, welches aus vielen heterogenen Subsystemen besteht. Es benötigt eine leistungsfähige Kommunikationsplattform, um alle relevanten Datenströme zu übertragen. Diese umfassen nicht nur 3D Videoinformation sondern auch Daten für Audio, Synchronisation, Benachrichtigungen, Interaktion, 3D Szenengeometrie und andere. Eine detaillierte Beschreibung des blue-c Kommunikationssystems kann in Lamboray et al. [4] oder in der Dissertation von Lamboray [5] gefunden werden.

Neben der 3D Videoinformation unterscheiden wir zwischen den folgenden Datentypen:

- *Bulk Data* umfassen grosse Datenmengen der Anwendung oder Szenenbeschreibung. Sie werden typischerweise bei Verbindungsaufbau einmal übertragen und nur sporadisch aktualisiert.
- *Event und Message-basierte Kommunikation*: Dies sind im Allgemeinen vom Benutzer initiierte Änderungen der aktuellen virtuellen Szene.
- *Echtzeit-Audio*: Diese ermöglichen die Sprachkommunikation in blue-c und sind sehr empfindlich gegenüber systembedingten Verzögerungen.
- *Tracking und Systeminformation*: Dies sind Daten zur Ressourcenverwaltung sowie Tracking-Daten zur dynamischen Kamerakontrolle.

Die Architektur des für blue-c entworfenen Kommunikationssystems ist in Abb. 13 dargestellt. Grundsätzlich galt es, die Forderungen nach zu-

verlässigen Services sowie ein fortschrittliches Programmiermodell, wie es z. B. von CORBA zur Verfügung gestellt wird, mit den Echtzeitanforderungen der 3D Videokommunikation in Einklang zu bringen. Zu diesem Zweck unterscheidet das Kommunikationssystem zwischen einem *Datenübertragungs-API* sowie einem *Service-API*.

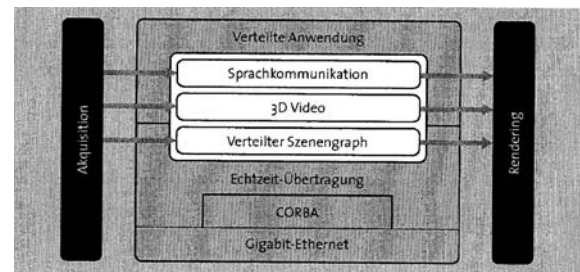


Abb. 13 Architektur der blue-c Kommunikationsstufe

Das Datenübertragungs-API des Kommunikationssystems stellt Kanalschnittstellen zum Versenden und Empfangen von Daten zur Verfügung. Die verteilten Anwendungsdaten sind in *Nachrichten-Objekte* organisiert, welche ihre Nutzdaten über Nachrichten in einen Übertragungspuffer schreiben. Der Empfänger analysiert den Übertragungspuffer,

DOI 10.1007/s00287-005-0057-5
© Springer-Verlag 2005

ETH Zürich
Institute of Computational Science
Haldeneggsteig 4
8092 Zürich
E-mail: grossm@inf.ethz.ch

die Nachrichten werden dekodiert und ein Callback-Mechanismus informiert die Anwendung über die angekommene Nachricht. Flusskontrolle und erneutes Übertragen stehen nur für den Übertragungspuffer zur Verfügung. Ein solcher Puffer kann sowohl Nachrichten nur eines bestimmten Typs als auch Nachrichten verschiedener Typen beinhalten. Lange Nachrichten werden zerlegt und über mehrere Puffer verteilt. Alle Puffer werden über Speicher-Pools behandelt, so dass die Speicherzuordnung nur während des Systemstarts erfolgen muss.

Die Echtzeitdaten werden über einen Kommunikationskanal übertragen, welcher aus einem Vorwärtskanal für Nutzdaten sowie einem Rückwärtskanal für Kontrolldaten besteht. Kontrolldaten beinhalten z. B. die aktuelle Kameraposition, Paketverlustraten, Paketschwankungen etc. Dieser Ansatz erfolgt in Anlehnung an das RTP/RTCP Protokoll, wobei RTP die unsichere Übertragung von Nutzdaten übernimmt und RTCP die periodische Kontrollinformation überträgt.

Das Service-API umfasst eine Reihe von Diensten, welche auf dem CORBA-Standard für verteilte Objektkommunikation beruhen. Der *Naming Service* dient zur Lokalisierung von verteilten Objekten. Der *Time Service* erlaubt es, synchronisierte Zeitmarken innerhalb des Netzwerks zu vergeben. Der *Notification Service* ermöglicht es, sporadische Ereignisse an interessierte Knoten zu verteilen. Das Verbindungsmanagement der Kommunikationskanäle basiert auf dem CORBA *Audio/Video Streaming Service*. Die aktuelle Implementation verwendet das *TAO/ACE Toolkit* (<http://www.cs.wustl.edu/~schmidt/TAO.html>).

Die Knoten eines einzelnen blue-c Portals sind über ein Standard 100 Mbps oder 1000 Mbps Ethernet Netzwerk verbunden, die beiden Portale hingegen ausschliesslich über Gigabit-Ethernet, wobei auch ATM-OC3 grundsätzlich unterstützt ist. Das System ermöglicht Voll-Duplex 3D Video, HiFi-Sprachkommunikation sowie eine Vielzahl von Benutzerinteraktionen.

Die im Abschnitt 3D Video-Verarbeitung beschriebenen 3D Video-Operatoren implementieren auf Stufe des Kommunikationssystems drei verschiedene Betriebsmodi für 3D Video in Anlehnung an den bekannten MPEG-Standard zur konventionellen Videoübertragung. Diese Modi umfassen einen I-Modus zur vollen Rekonstruktion, P- und R-Modi zur differentiellen Schätzung.

In der Praxis haben sich I- und P-Modi als weniger leistungsfähig erwiesen. Der R-Modus stellt einen guten Kompromiss für unsere Bedürfnisse dar (siehe auch Abschnitt Resultate und Anwendungen).

blue-c API

Das für den Benutzer sichtbare blue-c API stellt eine Anwendungsentwicklungsumgebung zur Verfügung, welche einen flexiblen Zugriff auf alle besonderen Funktionen der blue-c zulässt. Insbesondere kann der Benutzer 3D Graphik, Sprache, Klangsynthese und -ausgabe, 3D Video, Szenenverteilung, Interaktionsgeräte u. ä. leicht und komfortabel in eine Anwendung einbinden. Die entsprechenden Subsysteme werden als Dienste zur Verfügung gestellt und durch den blue-c API Kern verwaltet. Eine detaillierte Darstellung des API ist in Naef et al. [10] oder in der Dissertation von Naef [12] zu finden.

Das API ist so ausgelegt, dass es eine beliebige Anzahl von Portalen mit unterschiedlichen Hard- und Softwarekonfigurationen unterstützt. Einen konzeptionellen Überblick liefert Abb. 14.

Der Szenengraph, welcher die komplette virtuelle Welt repräsentiert, basiert auf dem SGI OpenGL Performer Toolkit, welches um Methoden zur Verteilung und Synchronisation erweitert wurde. Performer unterstützt mehrere synchronisierte Rendering-Pipes sowie Multiprozess-Rendering mit hohem Graphik-Durchsatz.

Der Szenengraph wurde um spezielle Knoten erweitert, welche die transparente Verwendung von 3D Video-Objekten erlauben. Sie sind den konventionellen Objektknoten gleichgestellt und unterstützen Transformationen, Culling und Okklusionsbehandlung. Die 3D Videofragmente werden in einem Callback mit optimiertem OpenGL Code darge-

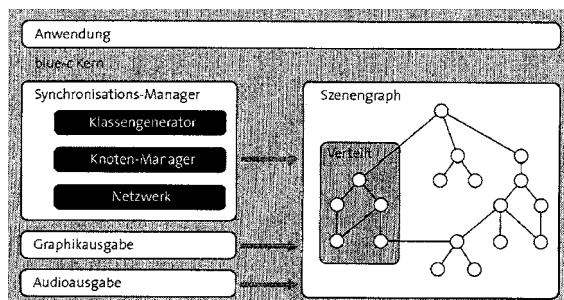


Abb. 14 Komponenten des blue-c API

stellt. Das eigentliche Rendering erfolgt über einen Algorithmus, welcher auf punktbasierte Graphik spezialisiert ist [23].

Ebenso sind Audio-Knoten Teil der Szene und erlauben es, räumliche Sound-Events einzubinden. Aktive Knoten unterstützen Objektinteraktion sowie die Auslösung von Animationen. Konventionelle 2D Videoknoten sind sowohl für lokale Videodateien als auch für verteilte Kameras über spezielle Texturknoten implementiert. Alle Knoten einer Szene besitzen eine Schnittstelle, welche Nachbildung und Synchronisation über die einzelnen Standorte hinweg erlaubt. Wir erreichen kurze visuelle Antwortzeiten auf Objektmanipulationen in Verbindung mit Szenenkonsistenz durch ein *relaxed locking* Schema, bei dem kurzfristige Inkonsistenzen schnell wieder behoben werden. Dabei werden Änderungen lokal zugelassen, ohne dass der Benutzer Besitzer des Knotens ist. Die Änderungen werden allerdings erst versendet, wenn der Benutzer zum Besitzer des Knotens wird. Selten kann es vorkommen, dass ein Besitzer-Transfer verweigert wird. In diesem Fall werden die lokalen Änderungen automatisch zurückgesetzt. Dies hat sich in der Praxis sehr bewährt. Das Synchronisationssystem mit leicht angepasstem Kommunikations- und Locking-System wird ebenfalls fürs Cluster-Rendering eingesetzt.

Die blue-c nutzt die volle Leistungsfähigkeit der parallelen Hardware des Graphik-Servers. Die Decodierung und die Vorverarbeitung der 3D Videostreams erfolgt in einem getrennten Prozess. Ebenso besitzen die Audio-Verarbeitung, die 2D Video-Decodierung sowie der Tracker eigene Prozesse. Die Interprozesskommunikation ist mittels *message passing* implementiert, welches die Verteilung über ein PC-Cluster ermöglicht. Das API steht im Moment sowohl auf einer Shared-Memory Architektur als auch auf PC-Clustern zur Verfügung und unterstützt IRIX, Linux und Windows Betriebssysteme.

Resultate und Anwendungen

Im Laufe des Projektes hat die Projektgruppe diverse Beispielanwendungen aus den Bereichen Architektur, Maschinenbau und Unterhaltung realisiert. Diese illustrieren die Leistungsfähigkeit des Systems und dienen als Plattform für Untersuchungen hinsichtlich der Systemperformanz.

Systemleistung

Die beschriebene 3D Videopipeline rekonstruiert im Moment 3D-Videos bestehend aus ca. 25 000 Fragmenten bei 5 Bildern/s oder ca. 15 000 Fragmenten bei 8,9 Bildern/s. Die Bildwiederholrate ist durch die aktuelle Hardware, insbesondere die synchronisierten Firewire-Kameras, begrenzt. Der verwendete Rechner-Cluster enthält 1,8 GHz Intel Pentium4 Prozessoren.

Abbildung 15 zeigt die Bandbreitenverteilung eines typischen differentiellen 3D Videostroms, welcher mit fünf geometrieaktiven und drei texturaktiven Kameras generiert wurde. Das Zeitfenster beträgt dabei 60 s, die durchschnittliche Bandbreite 2,6 Mbps. Die Spitzen in der Bitrate entsprechen den Bewegungen der Person und der Veränderung von aktiven Kameras, welche wiederum stark von den Veränderungen am virtuellen Betrachtungspunkt abhängen. Die stärksten Spitzen werden hauptsächlich bei der Übergabe von Kameras gemessen. Die Update-Rate der Fragmente kann über benutzerdefinierte Schwellwerte gesteuert werden; damit kann die Bildqualität gegenüber dem Bandbreiten- und Rechenbedarf abgestimmt werden. INSERT und DELETE-Operatoren steuern 25% und 12% der Gesamtbitrate bei, die restliche Bandbreite wird von den UPDATE-Operatoren verbraucht. In der Beispielsequenz von Abb. 17 werden 50 000 Operationen pro Sekunde verarbeitet. 40% der Operationen sind entweder INSERTs oder DELETEs, die verbleibenden 60% sind UPDATEs. Der Rekonstruktionsprozess erzeugt doppelt so viele Farb- wie Positionsveränderungen. Detailliertere Leistungsanalysen zum differentiellen 3D Videostrom sind in [22] zu finden.

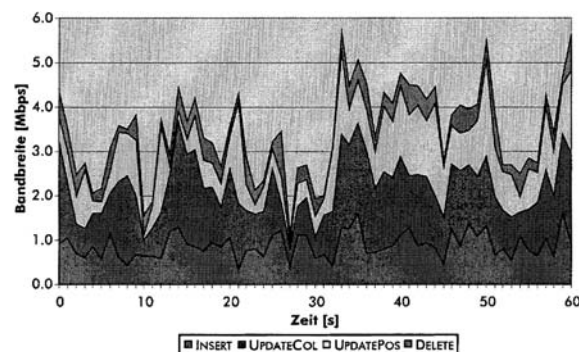


Abb. 15 Bitrate und Verteilung eines differentiellen 3D Videostroms über 60 Sekunden Aufnahmezeit

Die gesamte Systemlatenz von der 2D Akquisition bis zum Rendering beim Kommunikationspartner liegt zwischen 3 und 5 Zeitschritten. Die verschiedenen Betriebsmodi einer 3D Video-Übertragung werden nun in einem Experiment verglichen. Das 3D Videosystem wird in Echtzeit betrieben, allerdings werden die immer gleichen Bilder einer vorab aufgenommenen Videosequenz als Eingabedaten verwendet. Zusätzlich ermöglicht ein Simulationsmodul auf der Netzwerkebene Paketverluste nachzubilden. Als Qualitätsmerkmal für die 3D Videodarstellung verwenden wir die Anzahl der Punkt-Samples.

Abbildung 16 a) zeigt die Überlebensfunktion der Punkt-Samples für die unterschiedlichen Betriebsmodi. Hier zeigt sich der eigentliche Vorteil der redundanten Codierung. In diesem Versuch wurden unterschiedlich starke Paketverluste während der Übertragung simuliert. Die durchschnittliche Anzahl Punkt-Samples bleibt im R-Modus nahezu unverändert. Im P-Modus hingegen bewirken die Paketverluste grössere Veränderungen in der Auf-

lösung (siehe Abb. 16 a). Die im P-Modus nötige verlustfreie Übertragung bewirkt desweiteren unregelmässige Aufdatierungen beim Renderer (siehe Abb. 16 b). Die in dieser Simulation erzielten Ergebnisse konnten auch bei einer Internet-Übertragung eines Echtzeit 3D Videostroms von der ETH Zürich an die University of California in Davis beobachtet werden.

Wie schon im Abschnitt Kommunikation und Netzwerk erwähnt, kommt es im R-Modus zu temporären Inkonsistenzen, die aber über die Zeit ausgeglichen werden. Abbildung 17 a) zeigt ein Bild aus einer 3D Videoübertragung, in dem keine Fehler durch eine verlustbehaftete Netzwerkübertragung entstanden sind. Abbildung 17 b) zeigt ein ähnliches Bild mit fehlenden UPDATE- und DELETE-Operatoren. In Abb. 17 c) ist eine Reihe von INSERT-Operatoren verloren gegangen, dies zeigt sich an den fehlenden Punkt-Samples in der rechten Bildhälfte. Detailliertere Leistungsanalysen zum blue-c System sind in [5] zu finden.

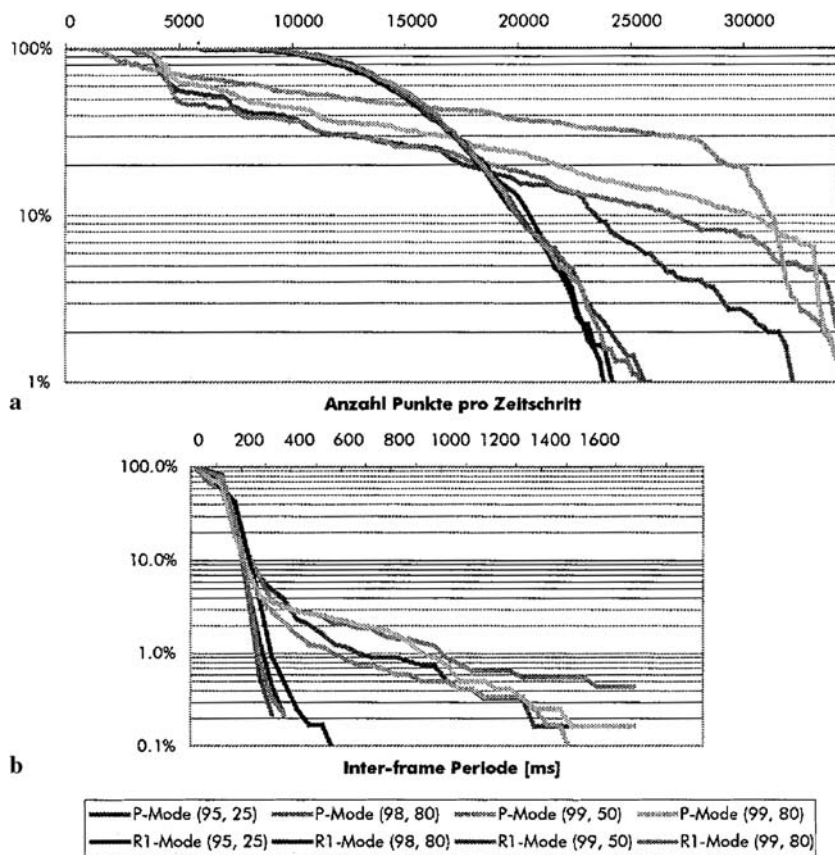


Abb. 16 Simuliert Bündelstörung:
a) Anzahl Punkte pro Frame;
b) Periode zwischen zwei Frames in Millisekunden

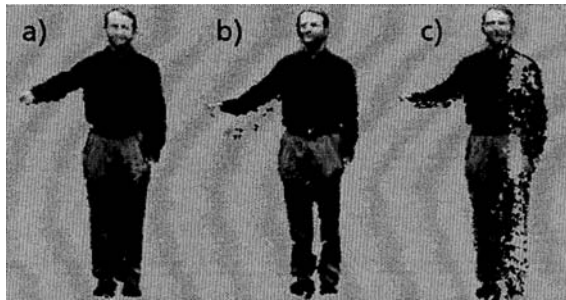


Abb. 17 Beispiel einer 3D Videoübertragung mit Visualisierung unterschiedlicher Artefakte, welche durch eine verlustbehaftete Übertragung entstehen: a) keine Paketverluste; b) Verlust von UPDATE- und DELETE-Operatoren; c) Verlust von INSERT-Operatoren

Bildqualität und Anwendungen

Zur Demonstration der Kernfunktionalität der blue-c implementierten wir zunächst einen „3D-Spiegel“⁴. Dieser erlaubt es dem Benutzer, ein dreidimensionales, stereoskopisches Bild seiner Selbst zu sehen und damit zu navigieren und zu interagieren. Der 3D Spiegel im blue-c Hauptportal ist in Abb. 18 illustriert und zeigt auch links unten das Bild der Rendering Engine, welches auf die Flüssigkristall-Projektionswände abgebildet wird.

Aufgrund der Stereoprojektionen ist es grundsätzlich nicht möglich, die visuelle Qualität der Projektion wiederzugeben, so wie sie vom Benutzer empfunden wird. Nichtsdestoweniger geben die Bilder einen Eindruck von der aktuellen Leistungsfähigkeit des Systems und zeigen dessen Grenzen auf. Es sei auch darauf hingewiesen, dass die Glanzlichter, welche in den Bildern vom Hauptportal auftreten, zu einem grossen Teil durch die Synchronisation der aktiven Brillen mit der Pro-



Abb. 18 3D Spiegel im blue-c Hauptportal mit entsprechendem projizierten Bild direkt von der Rendering Engine

jektion eliminiert werden. Zur Aufnahme ist das Projektionssystem zur besseren Illustration in den Monomode geschaltet worden.

Abbildung 19 gibt einen weiteren Blick in das Portal und illustriert die immersive Projektion. Der Anwendungshintergrund ist in diesem Fall Modedesign, wobei man sich vorstellt, dass ein räumlich getrenntes Model aktuelle Entwürfe präsentiert. Hierbei stellt die räumliche Dimension eine bedeutende Komponente zur zwischenmenschlichen Kommunikation sowie zur Erfassung von Form und Design dar. Eine detaillierte Untersuchung dieser Aspekte ist in der Dissertation von Lang [6] zu finden.

Abbildung 20 zeigt ein Bild einer bidirektionalen 3D Videokonferenz zwischen den zwei blue-c Portalen und illustriert die Benutzerinteraktion. Im kleinen Bildausschnitt links unten ist eine originale Kameraansicht des Nebenportals gezeigt. Das grosse Bild gibt wiederum einen Einblick in das

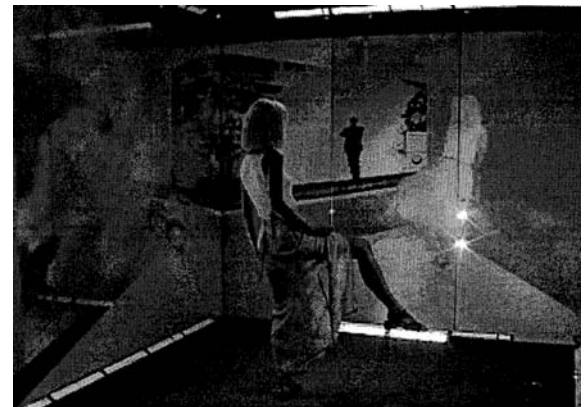


Abb. 19 3D Spiegel im blue-c Hauptportal



Abb. 20 Bi-direktionale 3D Video-Übertragung

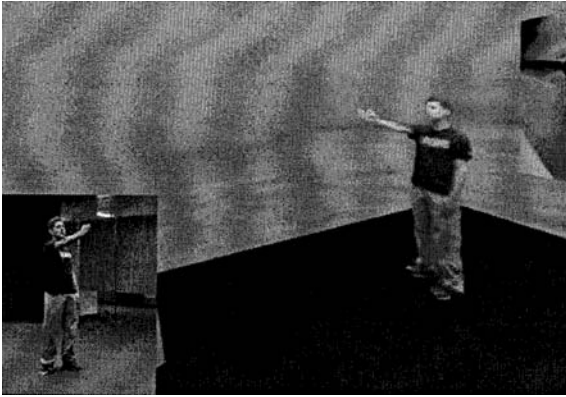


Abb. 21 Bildschirmphoto einer bidirektionalen 3D Video-Übertragung

Hauptportal und zeigt das 3D Videobild des Gegenübers, welches vom Benutzer, gesteuert durch eine 3D Maus, beliebig betrachtet werden kann.

Neben der Kameraauflösung ist die Qualität des 3D Videos von verschiedenen Faktoren abhängig: Erstens ist die 3D Rekonstruktion immer noch sehr berechnungsaufwändig und führt zu einer Limitierung der Auflösung des Fragmentstromes. Eine mögliche Erweiterung der Berechnungsknoten, welche die 3D Rekonstruktion durchführen, würde die Leistung signifikant verbessern. Abbildung 21 zeigt ein Bildschirmphoto vom Hauptportal einer 3D Video-Übertragung ausgehend vom Nebenportal und illustrieren die 3D Videoqualität in verschiedenen Auflösungsstufen.

Zusammenfassung und zukünftige Arbeiten

blue-c ist ein neuartiges Hard- und Softwaresystem, welches die Vorteile einer CAVETM-artigen Projektionsumgebung mit gleichzeitiger 3D Videoverarbeitung kombiniert und den Eindruck der vollständigen Immersion kreiert. blue-c ist ein bedeutender Schritt in Richtung wahrhafter Telepräsenz, also des Eindruckes „dort zu sein“. Wenngleich blue-c das weltweit erste System seiner Art darstellt, so müssen dennoch zahlreiche Limitationen in der Zukunft überwunden werden. Neben einer Verbesserung der Rekonstruktionsqualität soll blue-c insbesondere für mehrere Benutzer gleichzeitig nutzbar gemacht werden. Ferner bedarf es neuartiger haptischer Ausgabegeräte zur Rückkopplung taktiler Information.

Ein weiterer bedeutender Aspekt ist die Portabilität des Systems sowie seine Skalierbarkeit hinsichtlich der Anzahl der Projektoren und des Immersionsgrades. Unsere aktuellen Forschungen

konzentrieren sich auf die Bereitstellung von kleinen, portablen Einheiten, so genannter „Presence Bricks“, welche Projektor, mehrere Kameras sowie Lautsprecher und Mikrophone enthalten. Mittels der Methode des „unsichtbaren strukturierten Lichtes“ können wir Tiefenkarten hoher Qualität, für den Benutzer unsichtbar, erfassen und mit gleichzeitiger Projektion kombinieren [1]. Diese und weitere Aspekte in Bezug auf neuartige Interaktionsmöglichkeiten und kollaborative Anwendungen werden im Rahmen des Nachfolgeprojektes *blue-c-II* (<http://blue-c-II.ethz.ch>) erforscht.

Danksagung

Das blue-c Projekt wurde durch die ETH Zürich im Rahmen eines Polyprojektes, Nr. 0-23803-00, finanziert. Wir danken Maia Engeli, Ludger Hovestadt, Esther Koller-Meier, Andreas Kunz, Silke Lang, Markus Meier, Kuk Hwan Mieusset, Gerhard Schmitt, Oliver Staadt, Kai Strehlke, Tomas Svoboda, Andrew Vande Moere und Luc Van Gool für die Mitarbeit.

Literatur

1. Cotting, D., Naef, M., Gross, M., Fuchs, H.: Embedding imperceptible patterns into projected images for simultaneous acquisition and display. *Proceedings of the International Symposium on Mixed and Augmented Reality 2004 (ISMAR 2004)*, pp. 100–109, IEEE Computer Society Press 2004
2. Cruz-Neira, C., Sandin, D., Defanti, T.: Surround-screen projection-based virtual reality: The design and implementation of the cave. *Proceedings of SIGGRAPH 93*, pp. 135–142, ACM SIGGRAPH/Addison Wesley 1993
3. Gross, M., Würmlin, S., Naef, M., Lamboray, E., Spagno, C., Kunz, A., Koller-Meier, E., Svoboda, T., Van Gool, L., Lang, S., Strehlke, K., Vande Moere, A., Staadt, O.: blue-c: A spatially immersive display and 3D video portal for telepresence. *Proceedings of SIGGRAPH 2003*, pp. 819–827, ACM Press/ACM SIGGRAPH/New York 2003
4. Lamboray, E., Würmlin, S., Gross, M.: Real-time Streaming of Point-based 3D Video. *Proceedings of IEEE Virtual Reality Conference 2004 (VR 2004)*, pp. 91–98, IEEE Computer Society Press 2004
5. Lamboray, E.: A communication architecture for telepresence in virtual environments. *Ph.D. Thesis*, No. 15618, Department of Computer Science, ETH Zurich 2004
6. Lang, S.: The impact of video systems on architecture. *Ph.D. Thesis*, No. 15739, Department of Architecture, ETH Zurich 2004
7. Li, M., Magnor, M., Seidel, H.-P.: Hardware-accelerated rendering of photo hulls. *Computer Graphics Forum (Special issue on Eurographics 2004)*, vol. 23, no. 3, pp. 635–642, Blackwell Publishing Ltd. 2004
8. Matusik, W., Buehler, C., Mcmillan, L.: Polyhedral visual hulls for real-time rendering. *Proceedings of Twelfth Eurographics Workshop on Rendering*, pp. 115–125, Springer 2001
9. Matusik, W., Buehler, C., Raskar, R., Gortler, S., Mcmillan, L.: Image-based visual hulls. *Proceedings of SIGGRAPH 2000*, pp. 369–374, ACM Press/ACM SIGGRAPH/New York 2000
10. Naef, M., Staadt, O., Gross, M.: blue-c API: A multimedia and 3D video enhanced toolkit for collaborative virtual reality and telepresence. *Proceedings of VRCAI 04*, pp. 11–18, ACM Press 2004
11. Naef, M., Staadt, O., Gross, M.: Spatialized audio rendering for immersive virtual environments. *Proceedings of the ACM Symposium on Virtual Reality Software and Technology 2002*, pp. 65–72, ACM Press 2002
12. Naef, M.: blue-c API: Software support for collaborative virtual reality. *Ph.D. Thesis*, No. 15632, Department of Computer Science, ETH Zurich 2004

13. Pfister, H., Zwicker, M., van Baar, J., Gross, M.: Surfels: surface elements as rendering primitives. In *Proceedings of ACM SIGGRAPH 2000*, ACM Press/Addison-Wesley, New York, NY USA, pp. 335–342, 2000
14. Raskar, R., Welch, G., Cutts, M., Lake, A., Stesin, L., und Fuchs, H.: The office of the future: A unified approach to image-based modeling and spatially immersive displays. *Proceedings of SIGGRAPH 98*, pp. 179–188, ACM SIGGRAPH/Addison Wesley 1998
15. Spango, C.: Immersive virtual reality projektion system with simultaneous image acquisition using active projektion screens. Ph. D. Thesis, No. 15273, Department of Mechanical and Process Engineering, ETH Zurich 2004
16. Sadagic, A., Towles, H., Lanier, J., Fuchs, H., van Dam, A., Daniilidis, K., Mulligan, J., Holden, L., Zeleznik, B.: National tele-immersion initiative: Towards compelling tele-immersive collaborative environments. Presentation given at Medicine meets Virtual Reality 2001 Conference 2001
17. Spagno, C., Kunz, A.: Construction of a three-sided immersive telecollaboration system. *Proceedings of the IEEE Virtual Reality Conference 2003 (VR 2003)*, pp. 37–44, IEEE Computer Society Press 2003
18. Subramanian, S., Rajan, V., Keenan, D., Sandin, D., Defanti, T., Johnson, A.: A realistic video avatar system for networked virtual environments. *Proceedings of Seventh Annual Immersive Projection Technology Symposium, 2002*
19. Svoboda, T., Martinec, D., Pajdla, T.: A convenient multi-camera selfcalibration for virtual environments. *PRESENCE: Teleoperators and Virtual Environments*, 14(4), MIT Press 2005
20. Würmlin, S., Lamboray, E., Stadt, O., Gross, M.: 3D video recorder. *Proceedings of Pacific Graphics 2002*, pp. 325–334, IEEE Press 2002
21. Würmlin, S., Lamboray, E., Gross, M.: 3D video fragments: Dynamic point samples for real-time free-viewpoint video. *Computers and Graphics*, 28 (1), Special Issue on Coding, Compression and Streaming Techniques for 3D and Multimedia Data, pp. 3–14, Elsevier Ltd. 2004
22. Würmlin, S.: Dynamic point samples as primitive for free-viewpoint video. *Ph.D. Thesis*, No. 15643, Department of Computer Science, ETH Zurich 2004
23. Zwicker, M., Pfister, H., Vanbaar, J., Gross, M.: Surface splatting. *Proceedings of SIGGRAPH 2001*, pp. 371–378, ACM Press/ACM SIGGRAPH/New York 2001