Distributed Collaborative Construction in Mixed Reality

Christian Blank, Malte Eckhoff, Iwer Petersen, Raimund Wege and Birgit Wendholt
UAS Hamburg
Hamburg, Germany
Email: {first name}.{family name}@haw-hamburg.de

***Abstract*—Distributed collaboration, portable mobile applications, natural user interfaces and comprehensive systems have been identified as future research directions in recent reviews about mixed reality in construction. On the other hand, current research in the mixed reality field addresses movement and anthropometric realism as critical success factors for an immersive virtual environment. Advances in object tracking, online (human) 3D reconstruction and gestural interfaces accompanied by wearable mobile displays provide us with the technological base to contribute to the challenges in both areas. In this paper, we propose a comprehensive immersive environment for a distributed collaborative construction process in a mixed reality setup. Participants on remote sites, solely equipped with smart see-through glasses, are cooperating in the construction of a virtual 3D model combining real (tangibles) and virtual objects. We consider our solution to give most suitable support for a distributed collaborative construction task by increasing the immersion of the environment, i.e.: (1) creating the impression of real collaboration by mirroring the behavior of participants in a common virtual scene; (2) providing more natural interaction through freehand gestures; (3) increasing the physical experience of the user through wearable 3D displays and construction with tangibles.**

Keywords–Mixed Reality; Computer Supported Collaborative Work; Natural User Interaction.

I. Introduction

As has been shown by Rankohi et al. [1] and Chi et al. [2] mixed reality (MR) has been widely adopted in the construction field over the last decades. Hence, the same authors identify distributed collaboration, portable mobile applications and natural user interfaces (NUI) as future research topics. From the MR point of view, Dionisio et al. [3] consider the degree of realism in virtual environments as a relevant subject for future investigations. The closer gestural interfaces are to physical interactions - referred to as movement realism - and the higher the lifelikeness of virtual characters and humans - referred to as anthropometric realism - the higher the acceptance of the environment. Recent advances in 3D object tracking, 3D reconstruction, natural user interfaces and mobile MR devices, provide the means to bridge the gap between real and virtual collaboration tasks.

Given the latest technologies and research topics, we propose an environment for distributed collaborative construction where virtual and real processes converge because of (1) realistic images of participants in a common scene, (2) natural gestural interaction and (3) better physical experience through wearable displays and tangible interaction.

To be more specific, a scenario will illustrate the functionality of the environment. Participants on remote sites, solely equipped with optical see-through glasses, are cooperating in the construction of a virtual 3D model combining real (tangibles) and virtual objects. In the bottom cut-out of Figure 1 the virtual object is represented as transparent block (a partially completed marble track). The real object, a cube tagged with markers, can be attached to the virtual marble track and then gets replicated into a virtual piece of the track. Each participant will see the constructed model as overlay to the construction scene from an individual perspective. Manipulations of one participant will be transferred directly to all others. Since conflicting actions cannot be completely avoided in a distributed environment, we assume that all participants behave cooperatively. To provide best possible support for cooperation, each participant will see 3D reconstructed representatives of the others in the virtual scene, depicted as gray shaded characters in Figure 1. All manipulations on virtual objects and the replication of real into virtual objects are performed in a gestural manner. Equipped with mobile see-through glasses each participant will be able to inspect the virtual model from different perspectives by moving around in the scene.

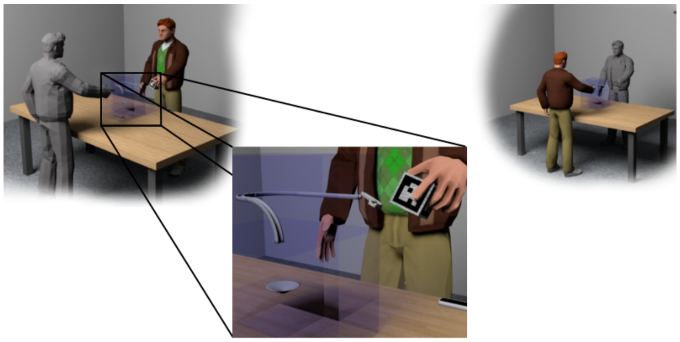


Figure 1. Overview of the proposed system. Two participants on two remote sites work collaboratively on a 3D model of a marble track.

In the remainder of this paper, section II briefly discusses the related work. Section III sketches the overall system design and introduces the system components. Section IV proposes a solution to ensure model consistency in a distributed construction process. Section V presents the components which are responsible for scene visualization. Section VI deals with the interaction techniques. Section VII sketches our solution for 3D online reconstruction of humans. Finally, section VIII will outline the major open work packages and future directions.

II. Related Work

A number of 3D MR construction environments have been developed in recent years. Salim [4] uses tangible building blocks and physical gestures to construct virtual urban landscapes on a 3D simulation table. Though already employing marker-less object tracking and object reconstruction, the solution is designed as a stationary, single user application. As opposed to [4], we employ marker-based object tracking with visible markers like in [5] or [6] for efficient identification and more reliable 3D pose detection. Though different proposals exist to estimate the object pose from non-coplanar feature

ISBN: 978-1-61208-382-7

points like in [7] and [8], we decided to implement an algorithm using coplanar feature points as described in [9] allowing for object pose calculation with only one recognized marker.

MirageTable [10] is an environment to combine the virtual and real world in a consistent virtual scene. It supports the construction task as combination of real and virtual objects and collaboration of users on a common model supplying 3D representations of the participants in the scene. It also allows for physically-realistic freehand gestures to manipulate virtual objects. Moreover, marker-less object tracking, object reconstruction and replicating real objects into virtual ones are contained. Though very close to our proposal, they follow a stationary approach with a stereoscopic projection. Here movement, fast on-line reconstruction of humans and dynamically changing perspectives are not considered. Since we propose a mobile setup, where participants are allowed to move around in the scene, there is a need for complete 3D models of all remote participants. Due to constant changes in perspectives, the reconstruction task needs to be performed in near real-time. Though Tong et al. [11] have shown the feasibility of using multiple consumer-grade depth cameras for reconstruction, their approach is too slow to create real-time dynamic meshes. As opposed to Alexiadis et al. [12], whose real-time reconstructed 3D models contain a high number of vertices, which is unsuitable for later streaming, in our solution, the data volume of the reconstruction process can be adapted in an early stage. This allows us to balance performance with mesh quality. As far as gestural interaction is concerned our work joins physical and interpreted gestures to achieve consistent device free interaction. For physical gestures we adapt the work of Song et al. [13] and Hilliges et al. [14]. For interpreted gestures we extend the template based approach of Kristensson et al. [15] for 2D gestures to 3D spatial interaction.

MixFab [16] is a MR environment for gesture-based construction of 3D objects in a stationary setting with a see-through display. Real objects are scanned by means of a depth sensor and can be combined with virtual ones. Manipulations range from joining real and virtual objects to deforming virtual by means of real objects. Having focus on mixed construction manipulations and gestures, MixFab does not support collaborative tasks, nor does it provide a mobile solution.

Mockup Builder [17] is a semi-immersive environment for freehand construction of 3D virtual models on a stereoscopic multi-touch table. The focus is on appropriate, convenient hand-gestures and thus, an excellent foundation for further development of our gestural interface.

III. System Design

The collaborative distributed environment consists of a couple of client instances. A client instance defines the environment which will run at all remote sites. Each instance consists of multiple components, which are communicating via a network middleware. Figure 2 shows the component dependencies, their attached sensor devices and the overall data flow. The reconstruction component is responsible for online 3D reconstruction of participants, whose results, the user meshes, are distributed to all other remote client instances for displaying purposes. The components tangible tracking and gesture recognition support the input side of the user interface. With tangible tracking, real objects can be incorporated in the construction process. Both, the construction logic and the scene visualization component are continuously informed about the actual positions of the real objects in the scene. The construction logic uses the position information to decide whether a real object's position is suitable for joining with the virtual construction model. The scene visualization tracks the real object's position by means of a virtual replicate. The gesture recognition component identifies gestures and informs about physical interactions with virtual objects. The construction logic component maps gestures onto model actions like joining or separating objects with/of the model. Physical interactions with virtual objects are appropriately reflected in the construction scene and model. The construction logic component ensures model consistency with respect to domain constraints and among all concurrent manipulations of client instances. The scene visualization component creates a consistent view of all output related data yielded by other components and renders the display data with respect to the view-ports, which are reported by the mobile display devices.

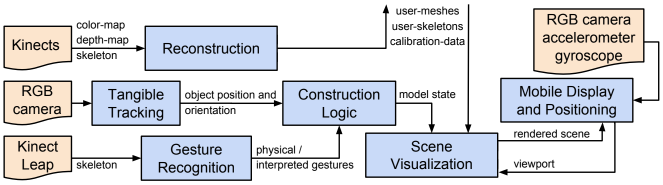


Figure 2. Client architecture

The backbone of component communication is the network middleware, which serves as an abstraction layer for the technical network. To achieve better scalability and extensibility, we have chosen a design of loosely coupled components and an event-driven and message-based communication style. To achieve location transparency, a service registry decouples network addresses from component services. Communication between client instances takes place in two ways: event-based, when actions of one client affect the underlying construction model and continuously, when user meshes are exchanged among instances.

IV. Construction Logic

The construction logic uses data from several components and controls the model construction based on domain-specific constraints. This module has to ensure model consistency and executes and resolves conflicting user actions in a concurrent distributed environment.

Concept: To ensure model consistency on a logical level, the very basic idea is to represent each entity as a building block with joints. Constraints are expressed in terms of joints, which specify criteria for valid connections. Only entities whose joints have matching criteria can connect to each other.

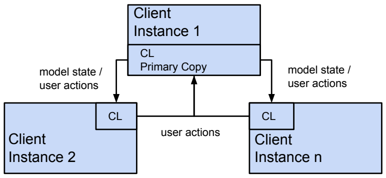


Figure 3. Distribution architecture. The construction logic (CL) component of each instance reports user actions to the primary copy, which synchronizes the model state and user actions among all client instances.

To enforce model consistency in a concurrent distributed environment in the first instance, user actions can only be performed on a primary copy (see Figure 3). Actions need to be executed in a strictly serialized and deterministic way to ensure consistent views for all users.

Current State: In an experimental setup, where the user can create a virtual marble track, the constraint-based construction approach has been validated successfully for different entity types and one unique constraint type. Results are shown in Figure 4 parts (b) and (c).

Future Work: The construction logic has to be further developed in several respects: (1) The logic itself has to be extended in order to cope with realistic domain models. (2) A solution for synchronizing replicated client-side model copies needs to be developed in order to enable consistent distributed construction.

V. Scene Visualization and Mobile Display and Positioning

Concept: The scene visualization component has to merge data from several components like (1) object positions from the tracking component, (2) model state from the construction logic and (3) user meshes from the reconstruction component. It will render the data into a consistent 3D scene and distribute the scene to all mobile displays of one client instance suitable for their individual view-ports.

Current State: Unity 3D is used as the engine for scene rendering and libGDX to display the rendered scene on Android-based mobile devices. All mobile devices permanently calculate their individual view-port and send it to the visualization component. The rotation matrix of a mobile module is calculated from an integrated accelerometer and gyroscope. A prototypical implementation for an Android tablet has been completed. The implementation for optical see-through glasses is currently under development. A more complicated task is to determine the head position. For now, a marker in the real scene represents the origin of the world space and gets tracked via a RGB camera of the mobile module.

Future Work: Using a marker to determine the head position, requires that users always look in the direction of the marker. A better solution will be global head-tracking for view-port-position calculation. This service will be integrated into the reconstruction component. Also, displaying user meshes in the scene is still an open task.

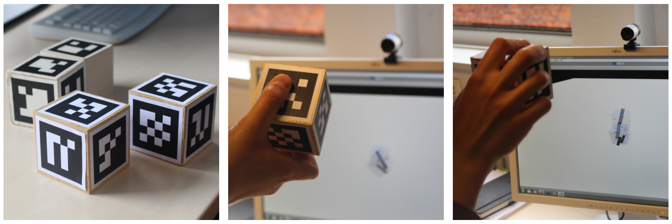
VI. Natural User Interface

Following Wachs et al. [18], who emphasize the role of natural user interfaces for intuitive and more natural interaction, we decided to exclusively use tangible and gestural interaction in our system. This section introduces the two subsystems responsible for tangible and gestural support.

A. Tangible Tracking

Unlike the common understanding of tangibles to serve as input controller we consider tangibles as real objects that become part of the construction model. Thus, tangible tracking for us means object tracking. For rapid prototyping purposes a marker-based solution has been implemented.

Current State: The marker-based approach uses cubes as representatives of domain entities, i.e., components of a marble track. Cubes carry unique, rotation-invariant markers on each side. A webcam is used as input device. Markers are continuously tracked in the image frames and marker positions are determined. Because of uniqueness and rotation invariance of the markers the basic orientation and the identity of objects can be determined. A coplanar POSIT algorithm [9] is used to estimate position and rotation of the marker relative to the camera. Finally, the actual position and rotation of the cube in the world system are calculated.



(a) Cubes represent marble track pieces. A cube carries unique markers, one marker for each side of a piece.

(b) Tracking: Moving cubes into the scene will create corresponding virtual marble track pieces. Virtual pieces follow the movement of the cubes.

(c) Constructing: Virtual pieces may join on valid connections which can be established through a gesture.

Figure 4. Constructing with tangibles.

The solution is capable of recognizing up to 5 unique objects in an area of 0.8 m in front of the camera with an update rate of 50 events per second. Occlusion is not handled.

B. Gesture Recognition

To support freehand 3D interaction with virtual objects a gesture recognition subsystem for interpreted as well as physical gestures is currently under development. Physical gestures are the virtual counterparts for interacting with objects in the real world, i.e., human movement has direct, realistic impact on virtual objects. Interpreted gestures are abstractions for movement patterns. These might be pointing gestures for menu item selection or object-related gestures like scaling virtual objects. The gesture recognition subsystem should provide a suitable abstraction layer to handle different kinds of input devices and multiple sensors. It should also be able to handle multiple users at the same time.

Concept: The gesture recognition subsystem consists of two major components: (1) Trame, a component for device abstraction in order to handle multiple sensor input and (2) the core recognition component for gesture detection, see Figure 5. Trame transforms sensor data into a common skeleton model. A controller of the core component dispatches the input to user related pipelines. These are processed in parallel, so that gestures of different users can be processed with interactive response times. Skeleton preprocessing will be used to smooth jitter, extract arm and hand positions, etc.

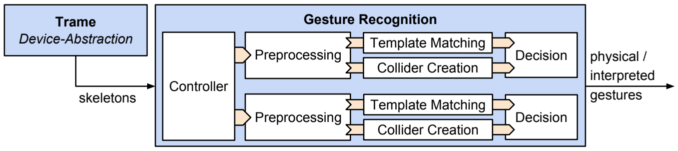
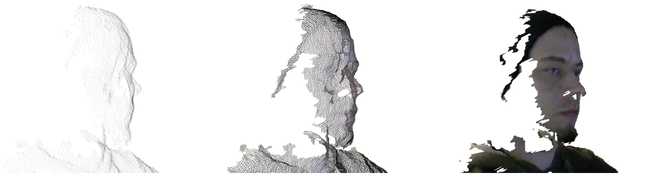


Figure 5. Overview of the gesture based interaction module with device abstraction and gesture recognition.

Template matching [15] enhanced with the observation of the third dimension and an extended set of input joints is responsible for detecting interpreted gestures. In parallel, a collider object, representing hand and arm movement, will be calculated in order to cope with physical gestures. In the decision step, interpreted gestures trigger corresponding events, which get distributed in the environment. In any case, a collider object will be provided for further processing.



(a) point cloud (b) wireframe (c) vertex colored triangle mesh

Figure 6. Reconstruction result using a single camera and organized fast mesh triangulation

Current State: Currently, Trame, the abstraction layer for sensor input, is implemented and supports Leap Motion and the Microsoft Kinect sensor.

Future Work: The implementation of the gesture recognition module for both kinds of gestures in a concurrent multi-user environment is one of the next goals to achieve. Afterwards usability studies need to be performed in order to verify the key assumption, that providing physical and interpreted gestures will give users an interface for interacting in a fast and natural way.

VII. User Reconstruction

When multiple users are operating in the MR environment, conflicting actions are imminent. We suppose that realistic virtual representatives of participants in a common scene will support cooperation since intentions of others might better be perceived. In technical terms, this means generating a closed textured polygon mesh for each user and visualizing it in the common 3D scene independently from the view-port.

Concept: As apposed to Alexiadis et al., who triangulate first and then combine multiple camera data - resulting in several mesh cleaning steps - this work proposes to first combine the data from multiple cameras and then to apply the mesh triangulation algorithm. Depending on the selected mesh triangulation algorithm some preparation steps may be necessary. The reconstruction pipeline therefore consists of a point processing step and a mesh triangulation step. The first step combines the data, and prepares the point cloud for the successive mesh triangulation. For example, a KD-tree of the point cloud is needed, when a moving-least-squares algorithm is used for triangulation. Multiple consumer-grade depth cameras are to be placed around a participant and calibrated to be able to transform the point cloud data into a common coordinate system.

Current State: Using a single depth camera a polygon mesh can be reconstructed at about 30 fps. The preparation step performs a background separation based on a thresholding approach only. The mesh is then triangulated with the organized fast mesh algorithm, which exploits point neighborhood relations known from the depth image. This naive meshing algorithm is not guaranteed to produce a closed mesh, as can be seen in Figure 6. While for a single camera this is a very efficient approach, it is not applicable to a point cloud assembled from data of several cameras.

Future Work: Future work includes the implementation of multi-camera management and a calibration method. Also needed is an evaluation of different processing pipelines for different meshing algorithms and related filtering steps in terms of speed and reconstruction quality. For example, moving-least-squares variants, greedy projection triangulation and marching cubes reconstruction are the next triangulation methods that will be evaluated. A solution for efficient mesh streaming is currently under investigation.

VIII. Conclusion and Future Work

In the preceding sections, we have outlined the architecture and components for a distributed collaborative construction environment. For each component, the current working state has been presented. Since we are at an early project stage, a number of open tasks have to be completed in the near future. In parallel, we are discussing realistic scenarios with manufacturing engineers and designers in order to verify our initial hypothesis, that virtual human representation and more natural interaction in conjunction with increased physical experience contribute to a better support for distributed construction. To end up with sound statements about our contribution to human computer interaction, we are planning a couple of comparative user studies with domain experts where we will investigate, whether (1) mixed construction has better acceptance and leads to better performance than pure virtual construction, (2) realistic 3D representations of participants better supports collaborative construction than employing self-animated avatars, (3) pure gestural and tangible interaction outranges more traditional interaction styles.

References

[1] S. Rankohi and L. Waugh, “Review and analysis of augmented reality literature for construction industry,” Visualization in Engineering, vol. 1, no. 1, 2013, p. 9.

[2] H.-L. Chi, S.-C. Kang, and X. Wang, “Research trends and opportunities of augmented reality applications in architecture, engineering, and construction,” Automation in Construction, vol. 33, no. 0, 2013, pp. 116 – 122, augmented Reality in Architecture, Engineering, and Construction.

[3] J. D. N. Dionisio, W. G. Burns, and R. Gilbert, “3d virtual worlds and the metaverse: Current status and future possibilities,” ACM Comput. Surv., vol. 45, no. 3, Jul. 2013, pp. 34:1–34:38.

[4] F. Salim, “Tangible 3d urban simulation table,” in Proceedings of the Symposium on Simulation for Architecture & Urban Design, ser. SimAUD ’14. San Diego, CA, USA: Society for Computer Simulation International, 2014, pp. 23:1–23:4.

[5] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, “Automatic generation and detection of highly reliable fiducial markers under occlusion,” Pattern Recognition, vol. 47, no. 6, 2014, pp. 2280–2292.

[6] M. Fiala, “Designing highly reliable fiducial markers,” Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 7, 2010, pp. 1317–1324.

[7] L.-J. Qin and F. Zhu, “A new method for pose estimation from line correspondences,” Acta Automatica Sinica, vol. 34, no. 2, 2008, pp. 130–134.

[8] D. F. DeMenthon and L. S. Davis, “Model-based object pose in 25 lines of code,” International Journal of Computer Vision, vol. 15, 1995, pp. 123–141.

[9] D. Oberkampf, D. F. DeMenthon, and L. S. Davis, “Iterative pose estimation using coplanar feature points,” Computer Vision and Image Understanding, vol. 63, no. 3, 1996, pp. 495–511.

[10] H. Benko, R. Jota, and A. Wilson, “Miragetable: Freehand interaction on a projected augmented reality tabletop,” in Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ser. CHI ’12. New York, NY, USA: ACM, 2012, pp. 199–208.

[11] J. Tong, J. Zhou, L. Liu, Z. Pan, and H. Yan, “Scanning 3d full human bodies using kinects,” IEEE Transactions on Visualization and Computer Graphics, vol. 18, no. 4, 2012, p. 643650.

[12] D. S. Alexiadis, D. Zarpalas, and P. Daras, “Real-time, full 3-d reconstruction of moving foreground objects from multiple consumer depth cameras,” IEEE Transactions on Multimedia, vol. 15, no. 2, 2013, pp. 339–358.

[13] P. Song, H. Yu, and S. Winkler, "Vision-based 3d finger interactions for mixed reality games with physics simulation," in Proceedings of The 7th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and Its Applications in Industry. ACM, 2008, p. 7.

[14] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson, "Holodesk: direct 3d interactions with a situated see-through display," in Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems. ACM, 2012, pp. 2421–2430.

[15] P. O. Kristensson, T. Nicholson, and A. Quigley, "Continuous recognition of one-handed and two-handed gestures using 3d full-body motion tracking sensors," in Proceedings of the 2012 ACM International Conference on Intelligent User Interfaces, ser. IUI '12. New York, NY, USA: ACM, 2012, pp. 89–92.

[16] C. Weichel, M. Lau, D. Kim, N. Villar, and H. W. Gellersen, "Mixfab: A mixed-reality environment for personal fabrication," in Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems, ser. CHI '14. New York, NY, USA: ACM, 2014, pp. 3855–3864.

[17] B. R. De AraúJo, G. Casiez, J. A. Jorge, and M. Hachet, "Special section on touching the 3rd dimension: Mockup builder: 3d modeling on and above the surface," Comput. Graph., vol. 37, no. 3, May 2013, pp. 165–178.

[18] J. P. Wachs, M. Kölsch, H. Stern, and Y. Edan, "Vision-based hand-gesture applications," Commun. ACM, vol. 54, no. 2, Feb. 2011, pp. 60–71. [Online]. Available: http://doi.acm.org/10.1145/1897816.1897838