Computers & Graphics 58 (2016) 53–65

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Issue on SMI 2016

Tangible 3D modeling of coherent and themed structures

Jeppe U. Walther *, J. Andreas Bærentzen, Henrik Aanæs

DTU Compute, Richard Petersens Plads, bygning 324, 2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Article history:
Received 14 March 2016
Received in revised form
7 May 2016
Accepted 9 May 2016
Available online 17 May 2016

Keywords:
TUI (Tangible User Interfaces)
3D shape modeling
Augmented reality
Depth cameras
Object tracking
Building blocks

ABSTRACT

We present CubeBuilder, a system for interactive, tangible 3D shape modeling. CubeBuilder allows the user to create a digital 3D model by placing physical, non-interlocking cubic blocks. These blocks may be placed in a completely arbitrary fashion and combined with other objects. In effect, this turns the task of 3D modeling into a playful activity that hardly requires any learning on the part of the user. The blocks are registered using a depth camera and entered into the cube graph where each block is a node and adjacent blocks are connected by edges. From the cube graph, we transform the initial cubes into coherent structures by generating smooth connection geometry for some edges of the graph. Based on an analysis of the cube graph, we identify subgraphs that match given graph templates. These subgraph templates map to predefined geometric refinements of the basic shape. This, in turn, allows the user to tangibly build structures of greater details than the blocks provide in and of themselves. We show a number of shapes that have been modeled by users and are indicative of the expressive power of the system. Furthermore, we demonstrate the scalability of the tangible interface which appears to be limited only by the number of blocks available.

1. Introduction

The idea of building edifices or structures with blocks, slabs or bricks is clearly ancient. Building with (small) blocks as a playful activity is more recent, but certainly much older than the digital computer. For instance, the Froebel blocks which inspired Frank Lloyd Wright [1] and had a big influence on kindergarten education, predate computer aided design by more than a hundred years [2]. Thus, we surmise that most people would find blocks familiar and would be able to start building with little instruction.

In this work, we present *CubeBuilder*. This system lets the user build with physical blocks, simultaneously acquiring a digital 3D model of the construction which is transformed into a contiguous, themed structure as the user builds.

Our interface is based on a Microsoft Kinect 2 [3,4] depth camera which allows us to detect where a block is placed. The CubeBuilder interface is highly scalable, since the addition of blocks does not increase the computational cost of detecting future blocks. Moreover, the Kinect may be moved during construction which greatly leverages the ability to place many blocks. Finally, we allow the building blocks to be placed freely and do not rely on interlocking building blocks.

Looking beyond the user interaction, we surmise that in many cases, building blocks will be a part of a playful activity where the blocks are placeholders that represent more complicated shapes. Our system allows users to actually create such shapes in two ways.

Firstly, the user is not required to place blocks in a completely regular fashion, but we create a graph of block connections: each face of each block may be connected to another adjacent block. For each such connection, we create a smooth *connection geometry* surface as shown in Fig. 6.

Secondly, we encode block configurations as templates for far more detailed structures based on a theme. When the user has placed an ensemble of blocks it is immediately interpreted according to this theme. Assuming a castle theme, the blocks will be embellished with arches, battlements, turrets etc. according to the structure of the brick ensemble.

To summarize, our contributions are

1. A tangible user interface to block modeling which has no inherent limitation on the number of blocks that a user can place.
2. The generation of connection geometry which allows for smooth, nonaxis-aligned structures to be created using our system.
3. A template based interpretation of the block structure, transforming a simple collection of blocks into a far more detailed shape.

* Corresponding author.
E-mail addresses: jeppewalther@gmail.com (J.U. Walther), janba@dtu.dk (J.A. Bærentzen), aanes@dtu.dk (H. Aanæs).

http://dx.doi.org/10.1016/j.cag.2016.05.004
0097-8493/

We envision that our system could be used in an augmented reality setting with creative and educational games in mind. The 3D models that users create could then coexist with interactive and animated characters.

2. Related work

The idea of creating *Tangible User Interfaces* (TUIs) with construction kits has previously been explored within the *constructive assembly* domain [5]. Early solutions were based on mechanical building blocks enriched with electronics [6–8] while more recent systems use depth cameras to register building blocks [9,10]. CubeBuilder is distinguished from many of these systems by allowing for motion of the depth camera and by leveraging the Kinect Fusion system [11,12] to maintain a separate volumetric model of the scene. This model makes it possible to completely decouple the representation of the designed block model from the computer vision system and facilitates a very scalable approach to block recognition. Since we can also move the sensor, the number of blocks can be very large. To the best of our knowledge, earlier systems do not allow for arbitrary (non-aligned) placement of blocks. The same is true about the application of a theme based interpretation of the structures with the exception of Anderson et al. [7] who use logical predicates to recognize blocks belonging to, say, a wall or a roof. This is used for embellishment. We posit that our subgraph matching combined with restrictions that are very similar to the predicates in [7] allows for better matching of structures that emerge when several blocks form a pattern.

If we consider systems for 3D modeling that are based on purely digital blocks, the game Minecraft [13] is without doubt the best known example. However, Minecraft strictly enforces a regular grid on the structures that one may build which is very different from our system where blocks may be placed freely. The same is true of the system LEGO Digital Designer (LDD) with which LEGO models may be built. Clearly, LDD impose certain structural constraints that are carried over from physical LEGO bricks. The block based modeling system by Leblanc et al. [14] is closer in spirit to our work: their system is not tangible, but users may place 3D blocks that are also joined to form a coherent mesh. Since the blocks are not physical, their system has few constraints on placement but does not offer the theme based reinterpretation of the block structure that our system does. However, Aliaga et al. [15] presented a system for visualization and interactive design of architectural models that applies architectural details to coarser structures in a procedurally defined facade model.

In an abstract sense, our system for detecting subgraphs that represent particular structures can be seen as an example of *inverse procedural modeling* [16] where the inverse modeling is greatly aided by the simplicity of block structures, and the goal is to replace the recognized structures by detailed models.

3. System overview

CubeBuilder allows the user to build virtual 3D models using a set of cube shaped physical building blocks. As the blocks are placed one by one (Fig. 1) to form larger structures, their positions and orientations are registered using input from a Kinect V2.

Users are presented with a real-time rendering of the digital model. This rendering does not necessarily correspond one to one with how cubes are placed in real-life. Instead details are added to the geometry in order to reflect what the user might imagine the cubes to represent. For instance, we can apply a castle theme as seen in Fig. 2.

CubeBuilder uses a SLAM [17] solution to track movement of the Kinect device in 6DoF. This allows the sensor to be moved which in turn allows for the creation of structures that are larger than what fits inside the Kinect's field of view. Knowing the position and orientation of the Kinect relative to the scene also allows us to create an augmented reality viewport where the themed geometry is rendered on top of a video stream from the RGB camera of the device.

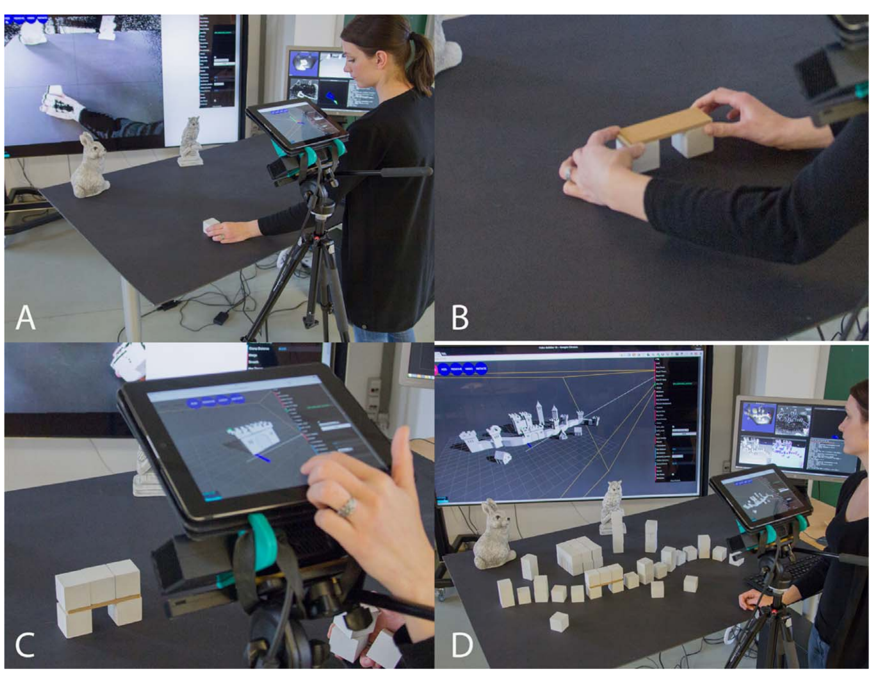

Fig. 1. With our system users can construct rich digital 3D models by combining physical blocks in front of a Kinect. A: The first cube is placed in the building area and registered via our setup. B: To make cubes levitate they can be combined with support objects for example in the form of building plates. C: While building, registered cubes are visualized in our user interface with applied themes. D: Structures larger than what can fit in the Kinects field of view can be achieved by moving the sensor while building.