Pacific Graphics (2019)
N. N, N. N, and N. N (Editors)

Collaborative 3D modeling system based on blockchain

Hunmin Park[1] and Sung-Eui Yoon[1]

[1] KAIST, Republic of Korea

Abstract

We propose a collaborative 3D modeling system, which is based on the blockchain technology. Our approach uses the blockchain to communicate with modeling tools and to provide them a decentralized database of the mesh modification history. This approach also provides a server-less version control system: users can commit their modifications to the blockchain and checkout others' modifications from the blockchain. As a result, our system enables users to do collaborative modeling without any central server.

CCS Concepts

• ***Computing methodologies*** → ***Shape modeling; Distributed algorithms;***

1. Introduction

With the development of the game and animation industries to meet the desire and expectation of consumers, we have to deal with large and complex 3D models. Therefore collaboration among many designers has become the common practice. Unfortunately, modeling tools like Blender do not support collaborative modeling yet due to its complexity. Instead, we can use a separate, version control system (VCS) such as Git. However, traditional VCS has a limitation that it depends on the central server(s) hosting the repository.

Departing from this centralized approach, the blockchain technology promises that we can share an immutable history of transactions without a central server. Based on the blockchain technology, we propose a new, collaborative 3D modeling system. Our approach has the following features:

- Decentralized, immutable, and Byzantine fault tolerant, since our approach is based on the blockchain.
- Modular, since the modeling part and the blockchain part of our approach are separated. So we can easily use various existing modeling tools in a project.

We think that our approach is useful for server-less collaborative modeling projects, especially community-based modeling projects. We expect our research to contribute to the modeling industry, and show that the blockchain can be used for various purposes even in graphics and modeling fields.

2. Related Work

Collaborative modeling. Doboš and Steed [DS12] presented an interactive visualization tool for mesh difference and conflict resolution. Also, a practical algorithm for diffing and merging polygonal meshes was presented [DP13]. Salvati et al. [SSTP15] proposed a real-time collaborative editing system of low-polygonal and subdivision meshes.

Blockchain-based revision control. There was a Kickstarter project [Ras], which tries to combine concepts of Git and blockchain. It attaches blockchain as a local proxy to Git for using Git without depending on Git hosting servers.

Blockchain-based database. A recent project attaches blockchain on a set of nodes where each node has its own MongoDB database to get the advantages of database and blockchain together [MMM*16].

While previous studies [DP13] [DS12] have focused on versioning meshes for collaboration modeling, our approach focuses on providing a distributed network that combines modeling tools with a decentralized database, allowing those prior studies to be used for server-less collaboration.

3. Collaborative Modeling using Blockchain

In this section, we first talk about our blockchain network, followed by various operations that our approach supports.

3.1. Network

Our idea for collaborative 3D modeling is to regard the modification of the mesh as a transaction and use the blockchain as a database of modification history. In the network, each node consists of one or modeling tools, e.g., Blender and Maya, and a blockchain

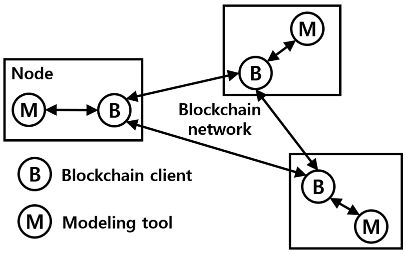

Figure 1: *This figure shows a blockchain network of our approach consisting of blockchain clients, each of which is associated with a modeling tool.*

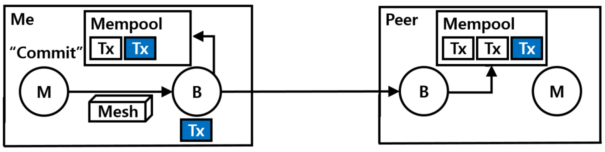

Figure 2: *Process of* commit*. The blockchain client creates the new transaction (shown in blue) using the committed mesh and broadcasts it. Each blockchain client appends the transaction to its mempool.*

client (Fig. 1). The blockchain client is responsible for distributing the transactions to the network, managing the blockchain, and delivering the modifications from other nodes to the modeling tool.

One may consider to integrate the modeling tool to the blockchain tightly or vice versa. Instead of this integration approach, we choose the current approach of separately having modeling tool and blockchain client, since this decoupled approach enables users to use any existing modeling tool as a plugin or to use multiple modeling tools together.

3.2. Operations

Our approach provides three operations: Commit, Mine, and Checkout.

Commit a mesh. When a user edits a mesh and requests *commit*, the modeling tool sends the mesh to the blockchain client. The blockchain client packs the mesh into a transaction. Instead of storing the whole mesh data, we store the difference between the current mesh and the prior mesh, which corresponds to the last transaction stored in the chain. We can retrieve the mesh later by iterating through the path from the root to the current transaction. This approach makes the transaction size a lot smaller than storing the whole mesh, so we can enable efficient collaborative modeling.

The blockchain client broadcasts the transaction to the peers, and each peer stores the transaction in temporary storage, *mempool*, for the transactions waiting for being included in a block (Fig. 2).

Mine a block. When a user requests *mine*, the blockchain client packs the transactions in the mempool and mines a new block. It updates its chain and broadcasts the block to its peers. When the peers receive the block, they update their chain (Fig. 3).

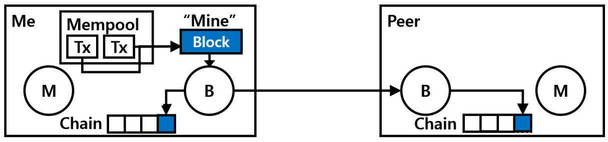

Figure 3: *Process of* mine*. The newly created block is shown in blue.*

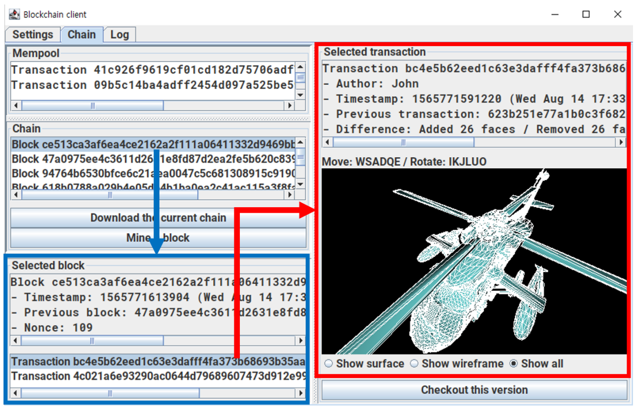

Figure 4: *Screenshot of the blockchain client. User can select a block from the chain (shown in blue) and select a transaction from the block (shown in red).*

Checkout a mesh. When a user selects a block from the chain and selects a transaction from the block and requests *checkout*, the blockchain client constructs the mesh corresponding to the transaction and sends the mesh to the modeling tool. This operation provides the user with a version control system. The user can download the latest commit or any other previous commit.

4. Implementation and Results

We have implemented a prototype for simulating our approach. The blockchain client is based on a simple PoW (Proof-of-Work) based blockchain written in Kotlin (Fig. 4). We wrote a plugin for Blender modeling tool for communicating with the client. The blockchain client provides a simple GUI for viewing and controlling the blockchain. The user can select a block from the chain and select a transaction from the block. When the user selects a transaction, the blockchain client provides a 3D preview of mesh, which corresponds to the transaction. Our implementation can be run on multiple localhost nodes in a single computer or multiple computers.

We have created a video for showing how our implementation works. Its link is at https://youtu.be/bEq464PwE4U. It shows how the operations work and how our server-less approach secures data from attacks and accidents. The code of our implementation is at https://github.com/Avantgarde95/C3DMB.

5. Acknowledgements

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7066317), and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2019R1A2C3002833)

References

[DP13] DENNING J. D., PELLACINI F.: Meshgit: Diffing and merging meshes for polygonal modeling. *ACM Trans. Graph. 32*, 4 (July 2013), 35:1–35:10. URL: http://doi.acm.org/10.1145/2461912.2461942, doi:10.1145/2461912.2461942. 1

[DS12] DOBOŠ J., STEED A.: 3d diff: An interactive approach to mesh differencing and conflict resolution. In *SIGGRAPH Asia 2012 Technical Briefs* (New York, NY, USA, 2012), SA '12, ACM, pp. 20:1–20:4. URL: http://doi.acm.org/10.1145/2407746.2407766, doi:10.1145/2407746.2407766. 1

[MMM*16] MCCONAGHY T., MARQUES R., MÜLLER A., ET AL.: *BigchainDB: A Scalable Blockchain Database*. Tech. rep., BigchainDB GmbH, jun 2016. 1

[Ras] RASHKOVSKII Y.: Gitchain. http://gitchain.org/. 1

[SSTP15] SALVATI G., SANTONI C., TIBALDO V., PELLACINI F.: Meshhisto: Collaborative modeling by sharing and retargeting editing histories. *ACM Trans. Graph. 34*, 6 (Oct. 2015), 205:1–205:10. URL: http://doi.acm.org/10.1145/2816795.2818110, doi:10.1145/2816795.2818110. 1