

Automated LEGO Assembly Construction by Interactive Selection from Multiple Optimization Techniques

Jae Woo Kim, Kyung-Kyu Kang, JiHyung Lee

Electronics and Telecommunications Research Institute, Korea

{jae_kim, kangk2, ijihyung}@etri.re.kr

Abstract—This paper presents a software environment to create a Lego model from input 3D mesh data. We added flexibility to the process of Lego assembly creation by providing user interaction. To achieve this, we implemented three different optimization algorithms. Users can interactively select a region of the input model data and then apply optimization algorithms with the feedback of stability evaluation.

Keywords— 3D Modeling, Lego Assembly Construction, Optimization

I. INTRODUCTION

Lego is one of the most popular toys in the world. It also has been used as a useful fast prototyping tool in mechanical engineering, CAD/CAM, structural analysis and other related areas. It is however difficult and time-consuming task to build a desired shape using LEGO pieces in a conventional way in which people repeat trying to construct a LEGO assembly structure by trials and errors. We therefore propose a novel interactive approach to efficiently build a LEGO assembly structures on computers [1][2].

Researchers have suggested several approaches with varying complexity of optimization algorithms and quality of resulting LEGO assembly structures. Each of the current approaches uses a single optimization technique to solve the problem. In general, there exists a trade-off between the processing time of the algorithm and the quality of the resulting LEGO structure [1][3-8]. We therefore suggest an approach to allow users to interactively select a local region of the 3D object and apply an appropriate optimization technique considering the importance and complexity of the model shape of current region. For example, for the complicated shape with higher importance, we can apply the optimization algorithm that takes more time but produces more convincing results.

The approach we suggest in this paper provides the following capabilities: The approach provides interaction in which a user can interactively select a region of interest. Once a region of interest has been selected, a user can select a set of heuristics that are used in optimization process to evaluate intermediate results and specify weight value for each heuristic. In a similar way, a user then selects an optimization technique she wants to apply in generating LEGO assembly structure for the current region and specify a set of parameters specific to the selected optimization technique. This research used the optimization techniques used in the approaches in previous research including greedy methods, simulated annealing, and evolutionary computation methods.

II. LEGO ASSEMBLY CONSTRUCTION

Current approaches to the automated Lego construction problem used greedy algorithms, local search algorithms, simulated annealing, cellular automata, and evolutionary algorithms to create Lego models [1][3][4][6]. Each approach however has pros and cons in terms of processing time and quality of the result. We can therefore selectively apply one of the multiple approaches to local regions of the input 3D model to maximize the performance of the solution.

The fundamental idea of our approaches is similar to the approaches suggested by reference [1][4]. Reference [1] firstly detect the "dangerous edges" and place Lego bricks to avoid the stability problems potentially caused by those edges. Secondly, they divide the remainder of each layer into regions. Finally, they adaptively select and apply a best approach to the current region among

the following approaches: (1) use of a library; (2) pattern filling and translating; (3) use of local search or simulated annealing. Reference [4] randomly selects regions in a layer and apply different types of crossover or mutation operators of evolutionary algorithms on the regions.

However reference [1] did not implement their approaches and use of a library or patterns is problematic because the patterns cannot be used for varying size of the input models. Reference [4] applied different genetic operators on current region instead of optimization techniques. Our goal is to improve the performance of the approach by adaptively applying several optimization techniques.

Our approach consists of the following steps. Firstly, we convert the input data to an legolized model. Secondly, we interactively apply one of the three different optimization techniques and test the stability of the intermediate results. The second step repeats until the whole Lego structure is constructed and then finally produce the output Lego structure and visualize it.

A. Legolized Model

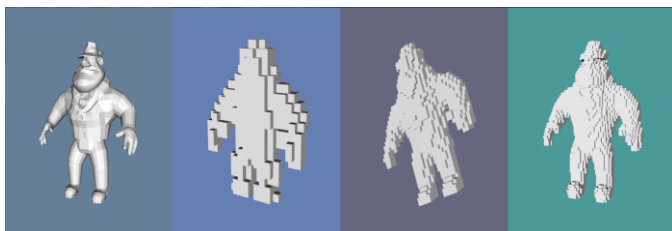


Figure 1 Legolized models in various levels of detail

Our approach inputs a 3D mesh model data and outputs the corresponding Lego model data. Before starting optimization process, we need to convert the input 3D mesh data into a legolized data that is a 3D matrix data whose element has the value one for the space occupied by a mesh data and zero for the empty space. We obtained the legolized data by running voxelization algorithm provided by VTK library. In this process, the level of detail can be specified by a user according to the user's intention on the quality of the resulting model.

B. Penalty Functions

There are three major performance criteria for the given problem as follows. First, the created Lego structure must be stable. Second, the processing time to create the Lego structure must be minimized.

Third, the number of Lego bricks used to create the Lego model must be minimized. The penalty functions for the optimization process was designed based on the first and the third criteria. To do this, we adopted the six heuristics for developing penalty functions suggested by reference [1]. The first three penalty functions described bellow used to guarantee the stability of the created Lego assembly: (1) High percentage of the area of each brick should be covered by other bricks from above or below, (2) Larger bricks must be preferred over small bricks, (3) Bricks in consecutive layers should have alternating directionalities

The other three heuristics for the penalty functions are concerned about the connectivity and stability at the boundaries of the 3D models: (4) A high percentage of the vertical boundaries of each brick should be covered by bricks in the bricks in the consecutive layers. (5) A brick must be placed such that the middle of the side should be at the boundary defined by the neighbouring bricks. (6) If a brick covers a vertical boundary in the previous layer, the middle of the brick must be aligned to the boundary.

C. User Interaction

Effective interfaces are provided to interactively select a region and optimization technique to apply and tune the parameters of the selected technique. Regions can be set by selecting a few consecutive layers or a local region in a layer. Optimization technique to apply is then selected and then the parameters are tuned. By using those interfaces, users can flexibly attempt to convert the model to Lego structure region by region. A possible use scenario is that a user select a relatively large region consisted of multiple layers and run a optimization that is relatively fast. Stability is then evaluated to see if the intermediate result is satisfactory. If the result is not satisfying, a user can make the region smaller or try to run a better optimization algorithm that requires longer running time. By doing in this way, we can run the faster algorithms for the simple shaped regions and the slower but better algorithms for the complicated shaped regions. The interaction strategy is also useful to investigate the relationship between the structure of a part of legolized model

and the optimization algorithm that is appropriate for the structure.

D. Optimization Techniques

We developed the following three algorithms: greedy algorithm, simulated annealing and evolutionary algorithm for our approach. Greedy algorithms firstly place the unit Lego bricks of size 1x1 to each voxel in legolized model. They secondly attempt to merge the unit bricks to replace each voxel with larger bricks. The bricks can be split if necessary to build more stable structure. Merge and split operations are repeated until the resulting model is stable [3][8].

Simulated annealing is a variant of the hill-climbing technique that computes all possible successor states from the current state and then selects the best successor. The well-know limitation of the approach is that it can easily converges to the local minimum instead of the global optimum [6].

At each iteration, simulated annealing algorithm considers a set of neighbouring states from the current state. It probabilistically compares between moving to new states and staying in the current state and then decide the new state that minimizes the energy. This process repeats until it finds a satisfactory solution

Evolutionary algorithms are very effective optimization technique for the problems whose optimal solution is hard to formalize. We adopted the genotype representation, crossover and mutation operators suggested by reference [4].

III. PRELIMINARY RESULTS

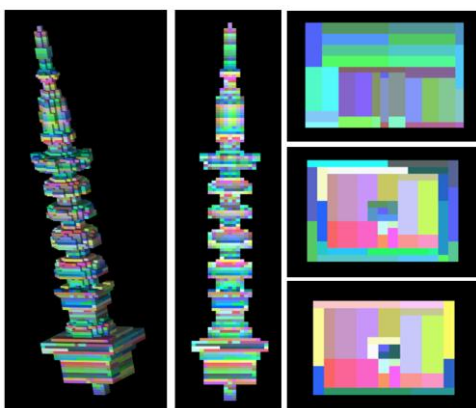


Figure 2 Preliminary result of Lego model construction: perspective view (left), side view (middle), and top views (right)

We developed the software system for automated LEGO assembly structure using Visualization Toolkit (VTK) libraries that provides a bunch of useful capabilities to handle 3D models, render the models, and interactively visualize the LEGO structure. This approach minimizes the processing time for optimization process in LEGO structure construction by applying relatively faster optimization algorithms for the region that has simpler structure while maintaining the overall quality of the resulting LEGO model. It also effectively reduces the chances to end up with an unsatisfactory result.

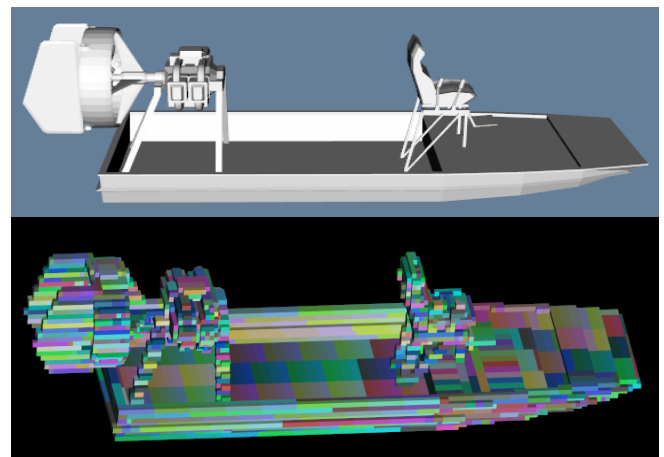


Figure 3 Lego model produced by our approach: an input 3D mesh model (above) and the corresponding Lego model (below)

Figure 3 shows an input 3D mesh model and the corresponding Lego model produced by our approach. Note that mostly larger bricks were placed on the layer that has large regular model data. This satisfies the heuristic 2.

Fig 4 shows the top views of consecutive layers of the Lego model. Note that horizontally and vertically aligned bricks were placed respectively. This improves the stability of the structure satisfying the heuristic 3.

Our observational analysis on the preliminary experimental data showed that our approach produced the results with better stability although the overall processing time to complete the Lego assembly construction was longer.

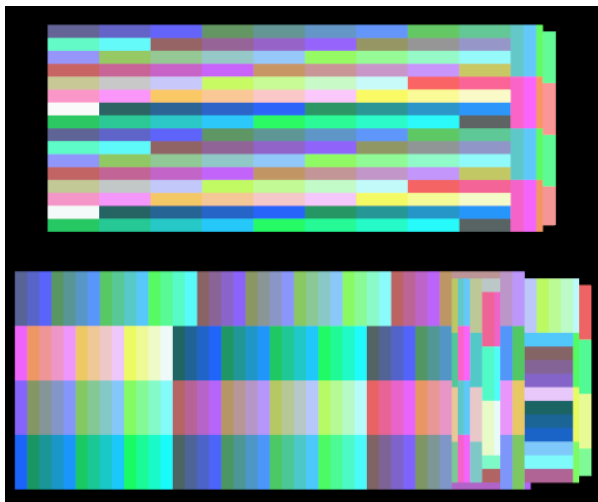


Figure 4 Two consecutive layers from the top view

IV. CONCLUSIONS AND FUTURE WORK

We have developed a software environment to create a Lego model from input 3D mesh data. We added flexibility to the process of Lego assembly creation by providing user interaction. To achieve this, we implemented three different optimization algorithms. Users can interactively select a region of the input model data and then apply optimization algorithms with the feedback of stability evaluation. This procedure improves the overall stability of the resulting Lego structure and further minimize the number of Lego bricks used to construct the Lego model. This is also similar to the conventional procedure in which humans build the Lego assemblies and therefore will be more entertaining.

Future work of this research includes optimal region selection to further automate the procedure. Sketch-based modelling and stereo vision techniques to acquire 3D models can be incorporated in this research to expand the applicability of the approach.

Larger data set of 3D mesh models must be included in experiments to rigorously validate the approach. The data set must include a number of 3D mesh data with a variety of different sizes, shapes, and complexities.

ACKNOWLEDGMENT

This research is supported by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program (Project ID: R2013040060).

REFERENCES

- [1] R. Gower, A. Heydtmann, and H. Petersen, "LEGO: Automated Model Construction," Jens Gravesen and Poul Hjorth, pp. 81-94, 1998.
- [2] O. Timcenko, "LEGO: How to build with LEGO," 32nd European Study Group with Industry Final Report, pp. 81-94, 1998. <http://www.maths-in-industry.org/past/ESGI/32/Report/ESGI32.ps>
- [3] S. Ono and A. Alexis, "Automatic generation of LEGO from the polygonal data," *International Workshop on Advanced Image Technology*, pp. 262-267, 2013.
- [4] P. Petrovic, "Solving the LEGO brick layout problem using evolutionary algorithms," Technical Report, Norwegian University of Science and Technology, 2001.
- [5] M. Peysakhov and W. Regli, "Using Assembly Representations to Enable Evolutionary Design of Lego Structures," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 17:155-68, 2003.
- [6] E. Smal, "Automated Brick Sculpture Construction," MS. Thesis, The University of Stellenbosch, 2008.
- [7] L. van Zijl and E. Smal, "Cellular automata with cell clustering," *Proceedings of Automata 2008 Workshop*, Bristol, UK, pp. 425-440, 2008.
- [8] R. Testuz, Y. Schwartzburg, and M. Pauly, "Automatic generation of constructable brick sculptures," *Eurographics 2013 Short Papers*, pp. 81-84, 2013.
- [9] B. Lambrecht, "Voxelization of boundary representations using oriented LEGO plates," University of California, Berkeley, 2006. http://lego.bladesign.org/LSculpt/lambrecht_legovoxels.pdf
- [10] L. Silva, V. Pamplona, and J. Comba Legolizer, "A real time system for modeling and rendering LEGO representations of boundary models," *XXII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI)*, pp. 17-23, 2009.
- [11] D. Winkler, "Automated brick layout," BrickFest, 2005. <http://www.brickshelf.com/gallery/happyfrosh/BrickFest2005/automatedbricklayout.pdf>



Jae Woo Kim is a senior researcher at Computer Graphics Research Section, Visual Content Research Department, Creative Content Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. He received his D.Sc. in Computer Science from the George Washington University and MS. in Computer Science and Bs. in Physics from Hankuk University of Foreign Studies. His research interest includes geometric modeling, computer animation, and visualization.



Kyung-Kyu Kang is a researcher at Computer Graphics Research Section, Visual Content Research Department, Creative Content Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. He received his Ph.D, MS, and BS in Media Engineering at Soongsil University in 2013, 2006, 2004. His interests include real-time rendering algorithms and physically-based simulation.



JiHyung Lee is a principal researcher at Computer Graphics Research Section, Visual Content Research Department, Creative Content Research Laboratory, Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea. He received his Ph.D in Computer Eegineering at Chungnam National University and MS in Computer Science at Korea University in 2011, 1996. His interest includes computer graphics and digital imaging.