LITERATURE REVIEW:
Parallel Beam Search for Functionality Partial Matching

Yanran Guan
School of Computer Science
Carleton University
Ottawa, Canada K1S 5B6
yanran.guan@carleton.ca

October 6, 2020

In the context of 3D shape modeling, functionality partial matching [4, 5] refers to the process of localizing the functionality analysis of 3D shapes from the category/object level down to the patch/part level, which can help today's cross-category shape modeling systems [24, 2, 5] that generate multi-functional shapes to evaluate the functionality of the generated shapes. In functionality partial matching, a shape is considered as a connected graph with each node denoting a part of the shape and each edge representing the connection between two parts. Thus, functional partial matching can be seen as a graph/tree traversal process that finds out all functionally plausible combinations of parts within a shape. Recently, a functionality partial matching method was proposed based on the beam search [4]. Beam search is a generalized breadth-first search (BFS) that only expands its search tree at the most "promising" nodes at each depth level, using a heuristic function. However, the current implementation of functionality partial matching uses a sequential beam search which is unoptimized and time-consuming. Thus, based on the idea of level synchronization [23, 1, 11], similar to a parallel BFS, we develop a parallel beam search to improve the running time of functionality partial matching.

1 Functionality Analysis in 3D Shape Modeling

Analyzing the functionality of 3D shapes [6] is a widely studied area in 3D shape modeling. Existing works on this topic have all focused on characterizing, comparing, or categorizing 3D objects based on their functions, which are typically inferred from their geometry and interactions with other objects or agents [10, 17, 9, 7, 16, 8]. In this work, we utilize the category functionality models developed by Hu et al. [7], of which each model computes a score for an input shape that describes how well the shape satisfies the functionality of a category. Furthermore, we localize the model to shape parts/part groups instead of directly applying it to an entire object, so as to enable functionality partial matching.

2 Beam Search and Its Applications

Beam search is a heuristic graph search algorithm and was first used in the HARPY speech recognition system by Bruce Lowerre [13]. The search tree of beam search is expanded in a breadth-first manner but it uses a heuristic function to select the nodes to be expanded at each level of its search tree. The set of selected nodes at each depth level of the search tree is called the beam. Due to its tractability and its efficiency in terms of both time and space complexity, beam search is usually applied in the case where the solution space of the graph is relatively large, e.g., in speech recognition systems [13, 3] and machine translation systems [20]. Recently, the sequence-to-sequence model in natural language processing [19] also uses a beam search for decoding. In our work, we use a heuristic function that cuts off the nodes with low scores at each step of tree expansion and preserves the most promising high score nodes.

3 Parallel BFS Algorithms

The BFS is a graph search algorithm is that explores the nodes of a graph level by level, where the level is defined as the set of leaf nodes that have the same depth to the root node of the search tree. Researches about parallel BFS algorithms cover its implementations for different architectures, e.g., for multi-cores [23, 1, 18, 11], for GPUs [12, 15, 21], and for distributed systems [14, 22]. The key idea of parallelizing a BFS is based on level synchronization [23, 1, 11]. Specifically, level synchronization makes sure that at each iteration only nodes having the same depth level are expanded to the next level. Moreover, the parallelism of a BFS can be further improved by using a multi-set data structure to replace the queue in traditional BFS algorithms [11]. Our method is based on the beam search, which can be seen as a special form of BFS. We also follow the idea of level synchronization to parallelize a beam search.

References

[1] David A. Bader and Kamesh Madduri. Designing multithreaded algorithms for breadth-first search and *st*-connectivity on the Cray MTA-2. In *Proceedings of the International Conference on Parallel Processing*, pages 523–530, 2006.

[2] Qiang Fu, Xiaowu Chen, Xiaoyu Su, and Hongbo Fu. Pose-inspired shape synthesis and functional hybrid. *IEEE Transactions on Visualization and Computer Graphics*, 23(12):2574–2585, 2017.

[3] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, 2013.

[4] Yanran Guan. 3D functionality analysis for shape modeling via partial matching. Master's thesis, Carleton University, 2020.

[5] Yanran Guan, Han Liu, Kun Liu, Kangxue Yin, Ruizhen Hu, Oliver van Kaick, Yan Zhang, Ersin Yumer, Nathan Carr, Radomir Mech, and Hao Zhang. FAME: 3D shape generation via functionality-aware model evolution. *IEEE Transactions on Visualization and Computer Graphics*, to appear.

[6] Ruizhen Hu, Manolis Savva, and Oliver van Kaick. Functionality representations and applications for shape analysis. *Computer Graphics Forum*, 37(2):603–624, 2018.

[7] Ruizhen Hu, Oliver van Kaick, Bojian Wu, Hui Huang, Ariel Shamir, and Hao Zhang. Learning how objects function via co-analysis of interactions. *ACM Transactions on Graphics*, 35(4):47:1–47:12, 2016.

[8] Ruizhen Hu, Zihao Yan, Jingwen Zhang, Oliver van Kaick, Ariel Shamir, Hao Zhang, and Hui Huang. Predictive and generative neural networks for object functionality. *ACM Transactions on Graphics*, 37(4):151:1–151:13, 2018.

[9] Ruizhen Hu, Chenyang Zhu, Oliver van Kaick, Ligang Liu, Ariel Shamir, and Hao Zhang. Interaction context (ICON): Towards a geometric functionality descriptor. *ACM Transactions on Graphics*, 34(4):83:1–83:12, 2015.

[10] Vladimir G. Kim, Siddhartha Chaudhuri, Leonidas Guibas, and Thomas Funkhouser. Shape2Pose: Human-centric shape analysis. *ACM Transactions on Graphics*, 33(4):120:1–120:12, 2014.

[11] Charles E. Leiserson and Tao B. Schardl. A work-efficient parallel breadth-first search algorithm (or how to cope with the nondeterminism of reducers). In *Proceedings of ACM Symposium on Parallelism in algorithms and architectures*, pages 303–314, 2010.

[12] Gu Liu, Hong An, Wenting Han, Xiaoqiang Li, Tao Sun, Wei Zhou, Xuechao Wei, and Xulong Tang. FlexBFS: A parallelism-aware implementation of breadth-first search on GPU. In *Proceedings of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 279–280, 2012.

[13] Bruce T. Lowerre. *The HARPY Speech Recognition System*. PhD thesis, Carnegie Mellon University, 1976.

[14] Huiwei Lu, Guangming Tan, Mingyu Chen, and Ninghui Sun. Reducing communication in parallel breadth-first search on distributed memory systems. In *Proceedings of the IEEE International Conference on Computational Science and Engineering*, pages 1261–1268, 2014.

[15] Enrico Mastrostefano and Massimo Bernaschi. Efficient breadth first search on multi-GPU systems. *Journal of Parallel and Distributed Computing*, 73(9):1292–1305, 2013.

[16] Sören Pirk, Vojtech Krs, Kaimo Hu, Suren Deepak Rajasekaran, Hao Kang, Yusuke Yoshiyasu, Bedrich Benes, and Leonidas J. Guibas. Understanding and exploiting object interaction landscapes. *ACM Transactions on Graphics*, 36(3):31:1–31:14, 2017.

[17] Manolis Savva, Angel X. Chang, Pat Hanrahan, Matthew Fisher, and Matthias Nießner. SceneGrok: Inferring action maps in 3D environments. *ACM Transactions on Graphics*, 33(6):212:1–212:10, 2014.

[18] Daniele Paolo Scarpazza, Oreste Villa, and Fabrizio Petrini. Efficient breadth-first search on the Cell/BE processor. *IEEE Transactions on Parallel and Distributed Systems*, 19(10):1381–1395, 2008.

[19] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112, 2014.

[20] Christoph Tillmann and Hermann Ney. Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Computational linguistics*, 29(1):97–133, 2003.

[21] Dominik Tödling, Martin Winter, and Markus Steinberger. Breadth-first search on dynamic graphs using dynamic parallelism on the GPU. In *Proceedings of the IEEE High Performance Extreme Computing Conference*, pages 1–7, 2019.

[22] Koji Ueno, Toyotaro Suzumura, Naoya Maruyama, Katsuki Fujisawa, and Satoshi Matsuoka. Efficient breadth-first search on massively parallel and distributed-memory machines. *Data Science and Engineering*, 2(1):22–35, 2017.

[23] Yang Zhang and Eric A. Hansen. Parallel breadth-first heuristic search on a shared-memory architecture. In *AAAI Workshop on Heuristic Search, Memory-Based Heuristics and Their Applications*, 2006.

[24] Youyi Zheng, Daniel Cohen-Or, and Niloy J. Mitra. Smart variations: Functional substructures for part compatibility. *Computer Graphics Forum*, 32(2):195–204, 2013.