Paper: Convex Covering

Jan “scymtym” Moringen

Yukari “Shinmera” Hafner

jmoringe@techfak.uni-bielefeld.de

shinmera@tymoon.eu

Shirakumo.org

Zürich, Switzerland

ABSTRACT

A

CCS CONCEPTS

• **Computing methodologies** → **Mesh geometry models**; **Computer graphics**; • **Applied computing** → Media arts.

KEYWORDS

Common Lisp, Convex Decomposition, Games, Video Games, Computer Graphics, Experience Report

ACM Reference Format:

Jan “scymtym” Moringen and Yukari “Shinmera” Hafner. 2023. Paper: Convex Covering. In *Proceedings of the 17th European Lisp Symposium (ELS’24)*. ACM, New York, NY, USA, 1 page.

1 INTRODUCTION

2 RELATED WORK

Liu et al.[2]’s work serves as the baseline for our implementation. Unfortunately their descriptions of the algorithm’s details aren’t entirely precise, making it difficult to reproduce their results exactly. We were also unable to find any publication of source code at all, let alone a working implementation.

Mamout et al.[3]’s work and their open implementation, “V-HACD”, provide a high-quality *approximate* convex decomposition algorithm. Their algorithm relies on a voxelisation step, which forces the source mesh into a watertight 2-manifold representation, and introduces deviations from the source mesh’s vertices. This can lead to noticeably different collision behaviour for terrain than the source mesh would produce. It can also easily drastically increase the total number of vertices compared to the source mesh, degrading collision performance. Their algorithm does allow tuning the complexity and number of produced hulls, but doing so requires human evaluation and there is no good general case behaviour.

Wei et al.[4] present a much improved method for approximate convex decomposition that is especially tuned for collision handling. However, their approach is extremely complex and difficult to implement, relying on many other algorithm implementations, such as 2D triangulation and monte-carlo tree search, which we would have had to reproduce in Lisp. Liu et al.’s original work only requires on an implementation of the Quickhull[1] algorithm.

3 ALGORITHM

4 EXTENSIONS

5 CONCLUSION

6 FURTHER WORK

7 ACKNOWLEDGEMENTS

REFERENCES

[1] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[2] Rong Liu, Hao Zhang, and James Busby. Convex hull covering of polygonal scenes for accurate collision detection in games. In *Graphics interface*, pages 203–210, 2008. URL https://www.cs.sfu.ca/~haoz/pubs/liu_zhang_gi08.pdf.

[3] Khaled Mamou, E Lengyel, and A Peters. Volumetric hierarchical approximate convex decomposition. In *Game Engine Gems 3*, pages 141–158. AK Peters, 2016.

[4] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics (TOG)*, 41(4):1–18, 2022. URL https://dl.acm.org/doi/pdf/10.1145/3528223.3530103.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

ELS’24, May 6–7 2024, Vienna, Austria

© 2024 Copyright held by the owner/author(s).

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.