On the Topology of Walkable Environments

B. Burton[1] A. Hillebrand[2] M. Löffler[2] S.Schleimer[3]
D. Thurston[4] S. Tillmann[5] W. van Toll[2]

Presented by J.L. Vermeulen[2]

[1]University of Queensland, Australia
[2]Dept. of Inform. and Computing Sciences, Utrecht University, the Netherlands
[3]University of Warwick - Coventry, Great Britain
[4]Indiana University - Bloomington, United States of America
[5]University of Sydney, Australia

European Workshop on Computational Geometry, 2018

Simulating crowds

Analysis of crowd disasters

- Hajj (2006, 2009)
- Love Parade (2010)

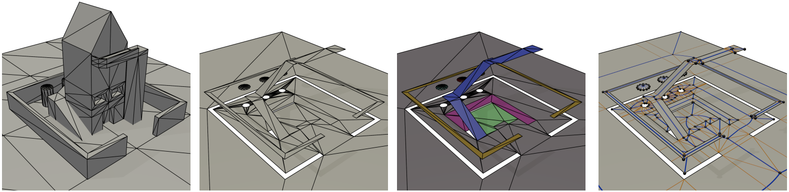
Improving safety

- Simulations of the Hajj
- Evacuations of concerts
- Grand Départ

Entertainment

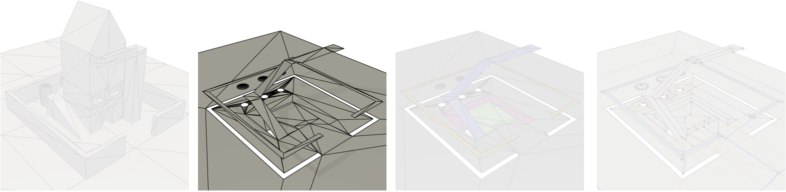
- Background crowds in games
- Crowds in movies

Preparing for crowd simulations



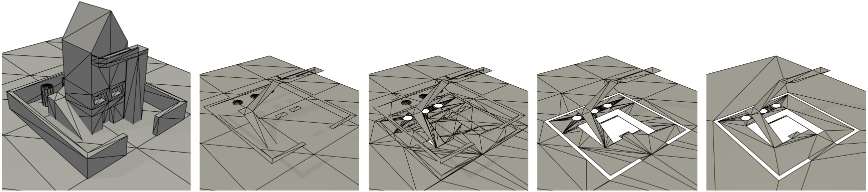
1. Obtain a 3D-model of a building;
2. Filter and repair to obtain the **walkable environment**;
3. Obtain a **multi-layered environment**;
4. Generate a **navigation mesh**.

Preparing for crowd simulations



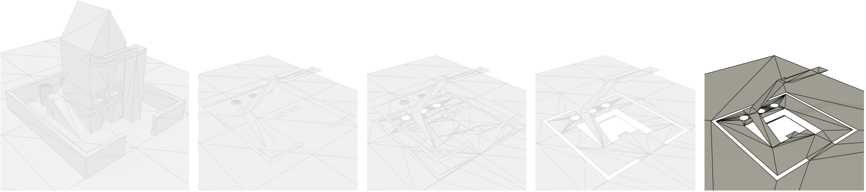
1. Obtain a 3D-model of a building;
2. Filter and repair to obtain the **walkable environment**;
3. Obtain a **multi-layered environment**;
4. Generate a **navigation mesh**.

Obtaining walkable environments



1. Obtain a 3D-model of a building;
2. Remove regions that are too steep;
3. Remove regions with not enough vertical clearance;
4. Remove regions that are too small;
5. Simplify the walkable environment.

Obtaining walkable environments



1. Obtain a 3D-model of a building;
2. Remove regions that are too steep;
3. Remove regions with not enough vertical clearance;
4. Remove regions that are too small;
5. Simplify the walkable environment.

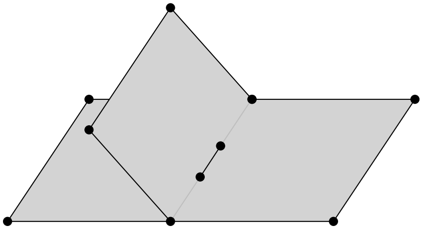
A small step in the process...

Goal: Simplify a walkable environment.

- Later, we will subdivide it into **layers** and treat each layer as a flat object
- Remove internal vertices
- **Re-triangulate** such that all diagonals are straight line segments (when viewed from above)
- These diagonals will be candidates for cutting the environment into layers

To re-triangulate properly, we need to understand the **geometry and topology** of a walkable environment.

Sloths

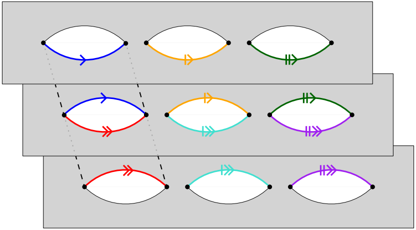


Definition (sloth)

A sloth is a compact surface continuously embedded in $\mathbb{R}^3$. Its boundary consists of **m** boundary vertices.

A sloth is **realistic** if the turning angle around any vertex is at most 2π.

Genus of sloths (1/2)



k layers, ℓ slits

Theorem

The genus of a sloth with m vertices is $O(m^2)$.

We use the **Euler characteristic** $\chi(\Sigma) = 2 - 2\text{genus}(\Sigma) - \#\partial\Sigma$.
We know $\chi(\Sigma) = \frac{1}{2\pi}\sum_i(\pi - \theta_i)$ (Gauss-Bonnet)

Genus of sloths (2/2)

Theorem

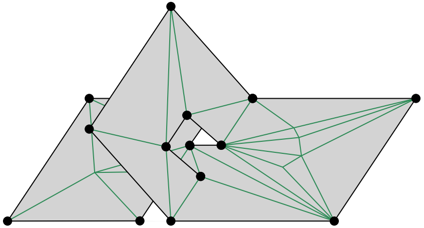
The genus of a **realistic** *sloth with m vertices is $O(m)$.*

This follows from the constant maximum turning angle around each vertex.

Corollary

Any **triangulation** *of a realistic sloth with m vertices will have $O(m)$ diagonals (just like for 2D polygons with holes).*

Walkable environments



Definition (Walkable environment)

A walkable environment is a **geometric representation of a realistic sloth** by triangles in $\mathbb{R}^3$ supported by **n** vertices, with the following restrictions:

- The angle between the ground plane and any triangle is $< 90^\circ$;
- The minimal vertical distance between any two triangles is non-zero.

Goal, revisited

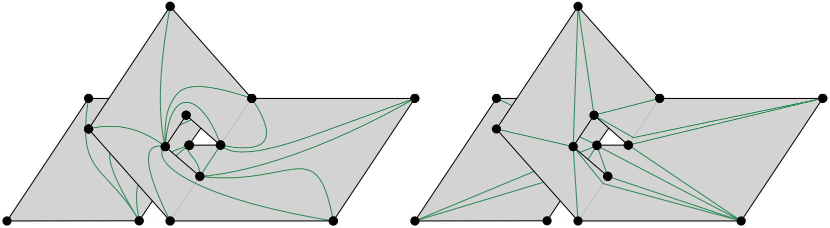
Goal: Simplify a walkable environment.

- Remove internal vertices
- **Re-triangulate** such that all diagonals are straight line segments (when viewed from above)

We now know that:

- The input has complexity $O(n)$;
- The output will have complexity $O(m)$;
- m can be much smaller than n.

Triangulations of sloths



Two types of triangulations of sloths:

Topological: Connect boundary vertices with arcs

Geometric: Connect boundary vertices with arcs that are **straight line segments** when projected onto $\mathbb{R}^2$

Triangulation of walkable environments

Theorem

A geometric triangulation of a walkable environment can be computed in $O(n + m \log m)$ *time.*

Algorithm is based on Lee and Preparata.
Instead of sweeping with a line, we sweep with a vertical plane.
All events of the algorithm happen at the m boundary vertices.

Number of connections in triangulation (1/2)

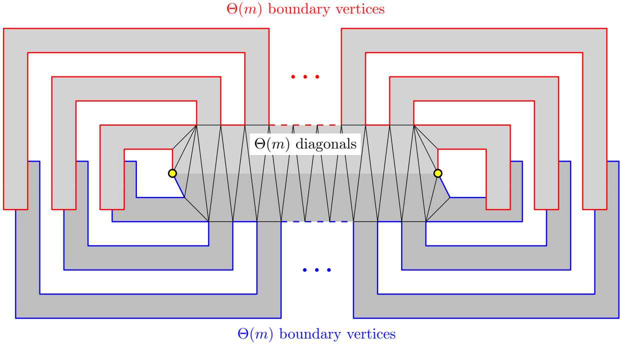
Recall: Diagonals of the triangulation will be candidates for cutting the environment into layers.
(A cut is then called a *connection* between layers.)

Question: Does every geometric triangulation yield a low number of connections?

Number of connections in triangulation (2/2)

Question: Does every geometric triangulation yield a low number of connections?

Answer: No, some can yield $\Theta(m)$ while others can yield $O(1)$.



Follow-up questions

Question: Can we always compute a ‘good’ triangulation that yields few connections?

Question: Given a sloth, can we efficiently find the lowest number of connections needed?

Question: Is there a relation between $\#\partial\Sigma$, genus and the treewidth of the dual graph of a geometric triangulation?

Question: Does a realistic sloth exist for which it may be a bad idea to restrict to a geometric triangulation?