Crowd simulation

Vojtěch Pröschl | Supervisor: prof. RNDr. Roman Barták, Ph.D.

(1) Crowd simulation

Animating large numbers of virtual characters is a recurring challenge in games and films. Manually animating each individual character is not feasible, leading to the development of crowd simulation techniques - **algorithms that automate the generation of realistic behavior for large groups**.

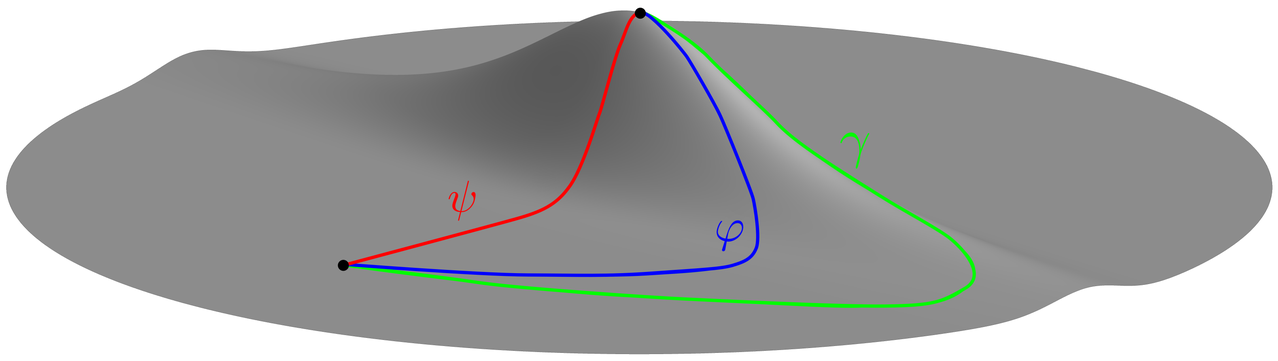
(2) Our contribution

We propose a framework to **generate motion curves for agents based on terrain** traversal difficulty. This difficulty is derived from the geometric properties of smooth curves and is controlled by a **user-defined cost functions**. We also explore a discretized version to ensure computational feasibility.

Continuous model

(3) Which path is the easiest to traverse?

The **shortest path is not always the easiest or most realistic**. The path ψ is the shortest in length, but traverses a steep and difficult ridge. Our cost-based model identifies the path φ as the optimal route, as it effectively **balances distance with traversal difficulty**.



(4) Curve length

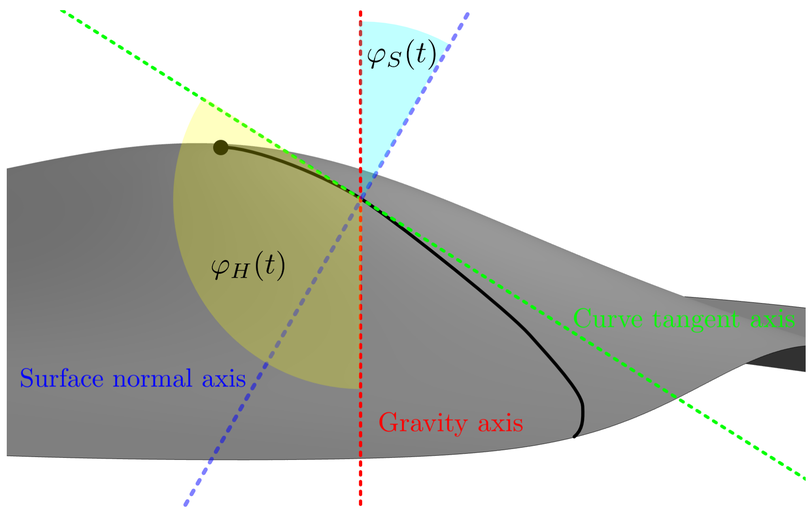
Standard curve length is computed by **integrating speed** along the path.

$$L(\varphi) := \int_{t_{\min}}^{t_{\max}} ||\varphi'(t)||\, dt$$

$L(\psi) \approx 2.58 \quad L(\varphi) \approx 3.84 \quad L(\gamma) \approx 5.96$

This measure incorrectly suggests ψ is the best path.

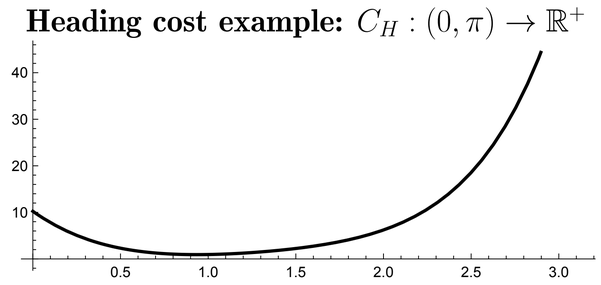
(5) Slope and heading angles



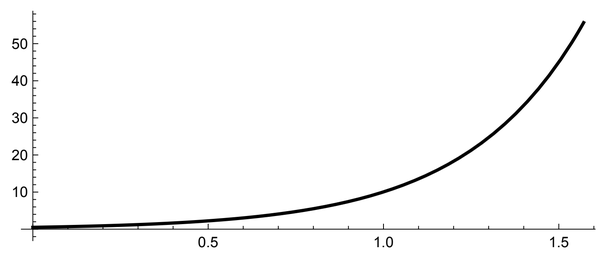
$\varphi_H(t)$ - **heading angle** at t $\quad \varphi_S(t)$ - **slope angle** at t

(6) Cost functions

Heading cost example: $C_H : (0, \pi) \to \mathbb{R}^+$



Slope cost example: $C_S : [0, \pi/2) \to \mathbb{R}^+$



(7) Joint cost function

Our joint cost function **integrates local difficulty**, defined by **slope and heading costs**, weighted by **speed** along the path.

$$C(\varphi) := \int_{t_{\min}}^{t_{\max}} (C_S(\varphi_S(t)) + C_H(\varphi_H(t))) \cdot ||\varphi'(t)||\, dt$$

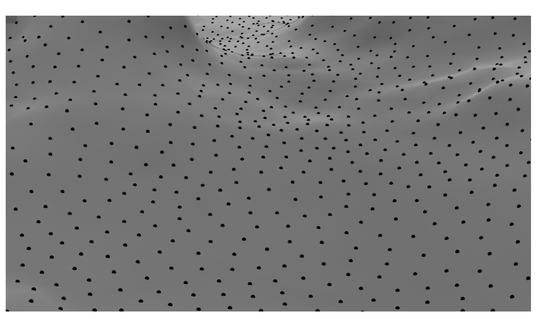
$C(\psi) \approx 49.49 \quad C(\varphi) \approx 32.36 \quad C(\gamma) \approx 35.24$

This measure identifies φ as the optimal path.

By altering the cost functions, **we can make the path γ the desired path**.

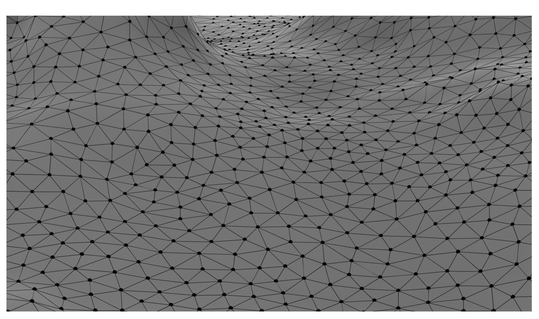
Discretization pipeline

(8) Sample surface



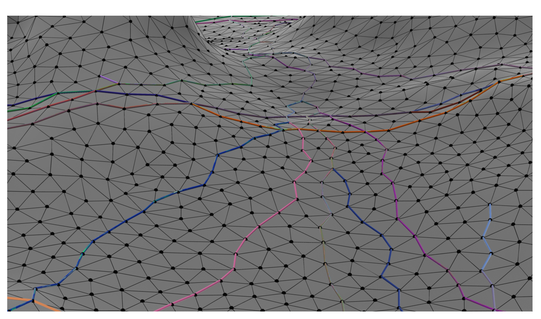
The continuous **terrain surface is sampled to create a set of vertices** for a graph. Methods such as **Poisson disk sampling** are suitable to ensure that vertices are **evenly distributed**, avoiding clustering.

(9) Generate edges



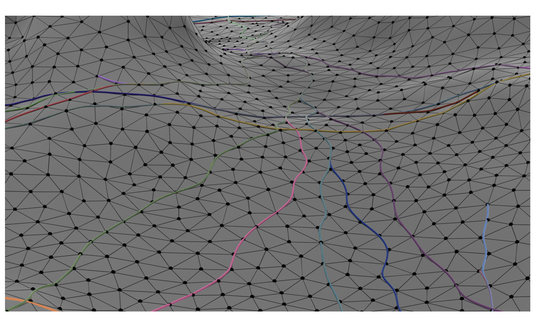
Vertices are connected using **Delaunay triangulation** to form a graph mesh. The cost of each edge is **computed by numerically** integrating our continuous cost function along the line segment connecting the two vertices.

(10) Find discrete paths



With a weighted graph, **standard pathfinding algorithms** such as the Dijkstra algorithm **can be used** to find the sequence of vertices that forms the path with the **lowest possible cost** from start to target for each agent.

(11) Interpolate paths



The **discrete path** (sequence of points) **is converted** back into a **smooth curve** for animation. **Catmull-Rom spline** can be used to generate curve that passes through every vertex of the discrete path, ensuring **natural-looking motion**.

(12) Evaluation

Comparison of difficulty measures

Distance-only cost favored shortest geometric paths but ignored traversal difficulty, **slope cost** avoided steep inclines but fails to modulate speed based on elevation gain, **heading cost** sometimes led agents along steep contours whereas the **joint cost** balanced these factors and produced the most human-like spatial and speed behavior.

Multi-agent behavior and collisions

On sparse terrains with few agents, collisions were minimal, in narrow corridors and regionally optimal routes, congestion and collisions occurred, indicating the need for dedicated avoidance strategies.

(13) Future work

Generalized surface models: Extend the framework to more general surface representations to handle complex terrains such as overhangs or caves.

New cost functions: Explore curvature, torsion, or spatial field–based costs.

Discretization guarantees: Develop methods with guarantees on how closely discrete paths approximate continuous optima.

Collision avoidance: Integrate explicit multi-agent avoidance and local resolution methods for crowded scenarios.

(14) Learn more