Collision detection using cross section approximations of meshes

Gábor FÁBIÁN*, Ákos RÁBELY
*Eötvös Loránd University, Faculty of Informatics
Department of Numerical Analysis

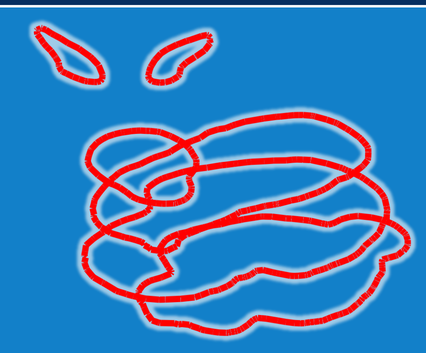
Abstract

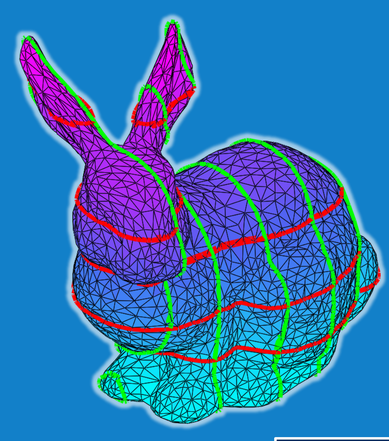
The key concepts of dynamic simulation of rigid bodies are collision detection and collision response, which allows the moving objects to change their momentum and angular momentum [3]. In computer graphics objects are mostly defined as polyhedra bounded by triangular surface elements (named triangle mesh), therefore the collision detection process in general can be considered as performing numerous triangle-triangle intersection tests. In some cases the colliding mesh contains so many triangles, that the exhausting triangle-triangle testing can not be done in real time. Thus various approximating methods are used to reduce the computation time. The common procedures are using bounding volume hierarchies (e.g. AABB) or space partitioning data structures (e.g. octree, k-d tree) and/or using a proxy, which is an approximation of the object containing much less triangles [1]. Unfortunately the reduced mesh (which contains sufficiently small number of triangles) sometimes is not a satisfactorily good approximation of the original one, therefore the dynamic simulation does not seem realistic.

In our research we try to approximate the model in an unusual way, which can be used in collision detection of an arbitrary polyhedron and a triangulated „ground" defined by a heightmap. The basic idea is to compute the cross sections of the mesh with some planes in orthogonal directions. If the mesh defines a self-intersectionless closed 2-manifold, then a cross section consists of simple polygonal lines [2]. If the mesh contains many triangles, the polygons may also have many vertices, consequently the approximation of these polygons is required. The coordinate functions of parameterizations of the polygonal lines are approximated by partial sums of Fourier-series. Using this approach we can show, that the collision detection problem is equivalent to testing intersection between two functions. The colliding point(s) can be found using a constrained minimization method, in our implementation the so-called simulated annealing algorithm was used [4]. Besides that, our method consists of multiple levels of preprocessing which helps us to reduce the computation time. This concludes that the time consuming minimization algorithm is called only if we can not exclude the collision of the mentioned cross section approximations.

Constructing cross section approximations

Slicing the mesh with planes perpendicular to the XZ-plane





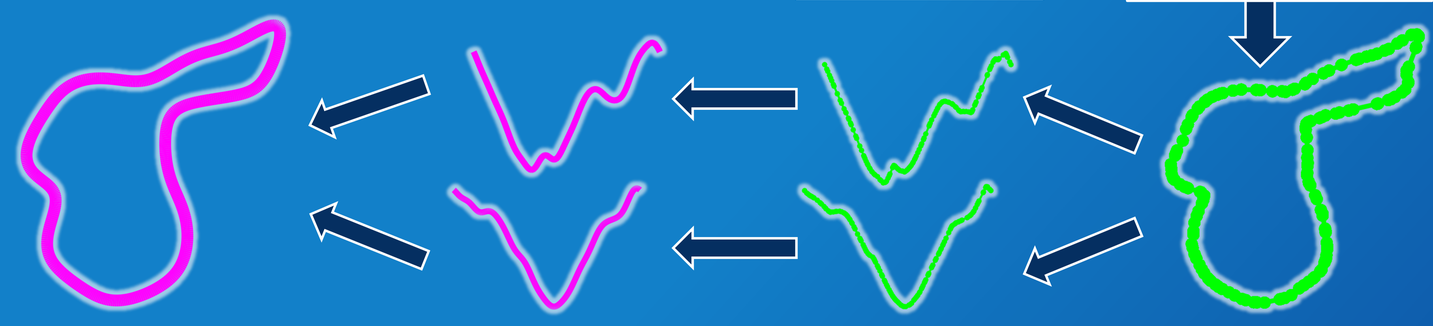
Slicing the mesh with planes perpendicular to the XY-plane

Closed planar curve

Partial Fourier sums

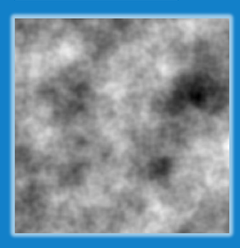
Piecewse linear parametrizations

Closed planar polygon

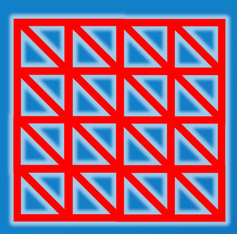


Ingredients

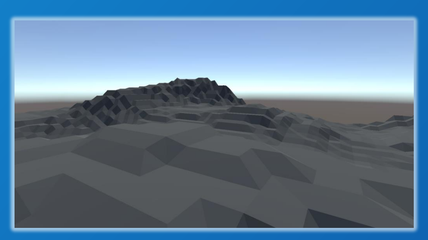
Heightmap



Regular grid



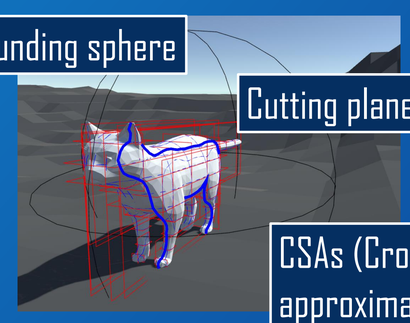
Generated terrain



Bounding sphere

Cutting planes + AABBs

CSAs (Cross section approximations)



Multilevel collision detection method

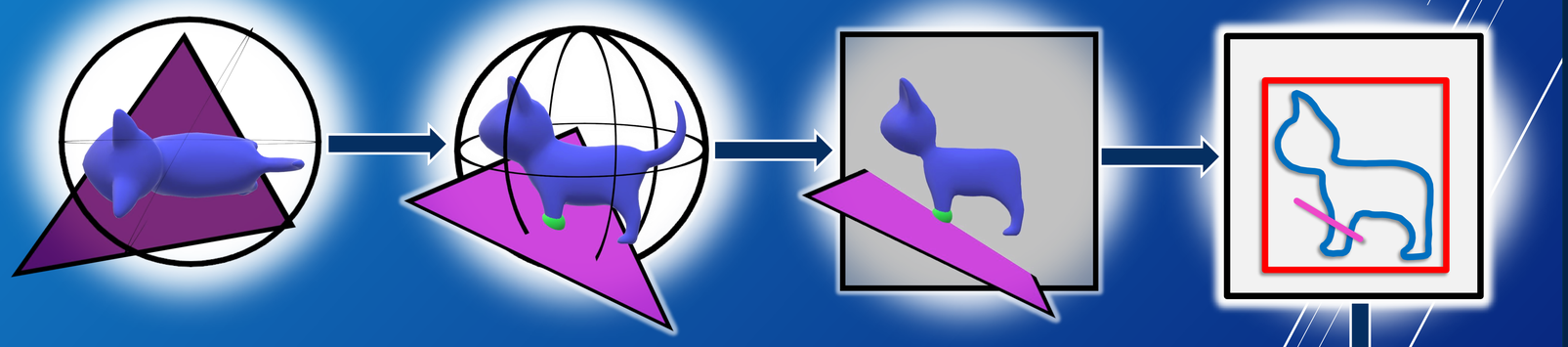
We assumed, that the terrain is generated by a regular triangle grid. The bounding sphere of the object, the cutting planes, the AABBs of the cross sections and the CSA curves are supposed to be precomputed. In each frame we do the following steps to determine the points of the CSA curves, that are in collision with an arbitrary triangle of the terrain.

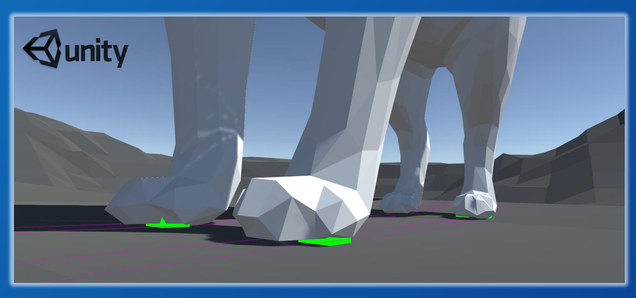
Filter the triangles

Triangle vs Bounding Sphere

Triangle vs Cutting plane

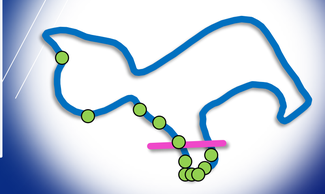
Line segment vs AABB





Simulated annealing is used to find the deepest point of the penetration. We are looking for the smallest y(t) value, but the objective function penalizes points that are not *below* the line segment.

Line segment vs CSA



Contact

Gábor Fábián
ELTE-FI Department of Numerical Analysis
1117 Budapest, Pázmány Péter sétány 1/C.
fabian@numanal.inf.elte.hu

References

1. Ericson, C., Real-Time Collision Detection, Morgan Kaufmann, (2005)
2. Fábián, G., Gergó, L., Fast algorithm to split and reconstruct triangular meshes, Studia Universitatis Babes-Bolyai Informatica, Vol. 59. Special Issue (1) (2014) 90-102.
3. Millington, I., Game Physics Engine Development, Morgan Kaufmann (2007)
4. Locatelli, M., Simulated annealing algorithms for continuous global optimization, Handbook of Global Optimization II, Kluwer Academic Publishers (2002)

© Poster Template by Genigraphics® 1.800.790.4001 www.genigraphics.com