Fast and Accurate Distance, Penetration, and Collision Queries Using Point-Sphere Trees and Distance Fields

Mikel Sagardia[*]
German Aerospace Center

Thomas Hulin[†]
German Aerospace Center

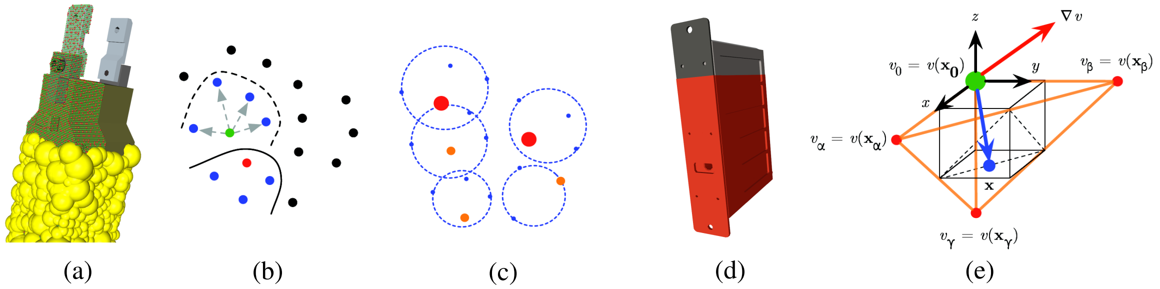
(a)
(b)
(c)
(d)
(e)
z
∇v
y
x
$v_0 = v(\mathbf{x_0})$
$v_\beta = v(\mathbf{x}_\beta)$
$v_\alpha = v(\mathbf{x}_\alpha)$
$v_\gamma = v(\mathbf{x}_\gamma)$
$\mathbf{x}$

Figure 1: *(a) Successive point tree levels and one sphere level of of a robotic gripper. (b) Clustering of the point tree. (c) Different point tree levels and minimal enclosing spheres of each cluster. (d) Voxelized satellite module. (e) Distance interpolation in a voxel octant.*

1 Introduction

Collision detection, force computation, and proximity queries are fundamental in interactive gaming, assembly simulations, or virtual prototyping. However, many available methods have to find a trade-off between the accuracy and the high computational speed required by haptics (1 kHz). [McNeely et al. 2006] presented the Voxmap-Pointshell (VPS) Algorithm, which enabled more reliable six-DoF haptic rendering between complex geometries than other approaches based on polygonal data structures. For each colliding object pair, this approach uses (*i*) a *voxelmap* or voxelized representation of one object and (*ii*) a *pointshell* or point-sampled representation of the other object (see Figure 2). In each cycle, the penetration of the points in the voxelized object is computed, which yields the collision force. [Barbič and James 2008] extended the VPS Algorithm to support deformable objects. This approach builds hierarchical data structures and distance fields that are updated during simulation as the objects deform.

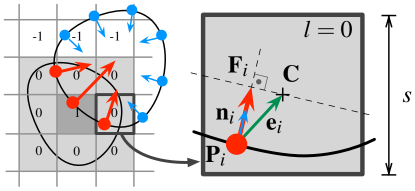

Figure 2: *Left: Voxelized and point-sampled objects in collision; Each voxel has its voxel layer value (l) related to its penetration in the voxelmap, and each point its inwards pointing normal vector ($\mathbf{n}_i$). Right: Single point ($\mathbf{P}_i$) force ($\mathbf{F}_i$) can be computed scaling the normal vector ($\mathbf{n}_i$) with its penetration in the voxelmap. The cross products of forces and points yield torques.*

We present a haptic rendering algorithm for rigid bodies based on the VPS Algorithm which also uses hierarchies and distance fields. Yet, our data structures are optimized for fast and accurate collision and proximity queries rather than for deformation simulations.

2 Our Approach

First, layered voxelmap and plain *point-soup* representations of objects are computed according to [Sagardia et al. 2008]. Then, real distance-field values (v) are stored in the voxels close to the surface and a point-sphere tree is built (down-top) upon the plain *point-soup*. In order to build our tree, neighbor points are organized in clusters (see Figure 1(b)). The point in the cluster which is closest to its center of mass belongs to the upper level in the tree, which is also clustered. Additionally, all points and children points within a cluster are enclosed with a minimal sphere (see Figure 1(c)).

After the offline generation of our structures, the online algorithm computes the penetration of the likely colliding points in the voxelmap, similarly as in Figure 2. Each haptic cycle, the uppermost cluster with the sphere that encloses all points is pushed to the query queue. The algorithm checks whether each popped cluster sphere is in collision; if so, the parent point of the cluster is checked for collision and children clusters are pushed to the queue. As shown in Figure 1(d), the floating-point (penetration) distance (v) of a point is interpolated after computing the local distance field gradient (∇v) in the voxel neighborhood (α, β, γ). Normal vectors scaled with the penetration yield collision forces (see Figure 2).

Figures 1(a) and 1(d) show examples of our data structures. Our algorithm is between 1.5 and 17 times faster than the tree-less approach in the classical *peg-in-hole* scenario and presents low aliasing artifacts. Future work will address the exploitation of distance queries and the uniformity in point distribution and clustering.

References

Barbič, J., and James, D. 2008. Six-dof haptic rendering of contact between geometrically complex reduced deformable models. *Haptics, IEEE Transactions on 1*, 1 (jan.-june), 39 –52.

McNeely, W. A., Puterbaugh, K. D., and Troy, J. J. 2006. Voxel-based 6-dof haptic rendering improvements. *Haptics-e: The Electronic Journal of Haptics Research 3*.

Sagardia, M., Hulin, T., Preusche, C., and Hirzinger, G. 2008. Improvements of the voxmap-pointshell algorithm - fast generation of haptic data-structures. In *53. IWK - TU Ilmenau*.

[*] e-mail: mikel.sagardia@dlr.de
[†] e-mail: thomas.hulin@dlr.de

SIGGRAPH 2013, July 21 – 25, 2013, Anaheim, California.
2013 Copyright held by the Owner/Author.
ACM 978-1-4503-2261-4/13/07