Edge collision detection in complex environment

T. Jund
LSIIT, UMR CNRS 7005
University of Strasbourg
France
thomas.jund@lsiit-cnrs.unistra.fr

D. Cazier
LSIIT, UMR CNRS 7005
University of Strasbourg
France
david.cazier@lsiit-cnrs.unistra.fr

J.-F. Dufourd
LSIIT, UMR CNRS 7005
University of Strasbourg
France
jean-francois.dufourd@lsiit-cnrs.unistra.fr

***Abstract*—We present in this paper a new approach to detect and precisely report the collisions and contacts between deformable moving bodies and their environment. The moving bodies are sampled as meshes whose vertices are followed up in a convex subdivision of the surrounding space. Particles are continuously spanned along the edges to detect collisions with cells of this subdivision.**

We report experimentations with shape matching based physical simulations and discuss performance of our method. We compare our approach with hierarchical ones.

I. INTRODUCTION

Continuous collision detection (CCD) for deformable moving bodies is a complex task in real-time surgical simulations. In this context, scalpels or catheters are moved inside a virtual patient body. Classical collision detection algorithms provide estimations of the interpenetrations volumes that are usually translated in the mechanical subsystem as penalties forces. That is not sufficient in this kind of simulations where the contact between the tool and the environment should be precisely acquired to provide the best visual and haptic feedback.

In this paper, we consider that the moving bodies lie in an environment modeled as a volumetric mesh. This mesh describes a set of convex polyhedrons tied by pairs, some faces of which are marked as impassable – the external boundary and possible internal obstructions. The moving bodies cannot get over vertices, edges or faces of the unreachable areas. The moving bodies are modeled as manifold meshes, but the faces of the bodies are not considered for the collision detection – i.e. intersections between faces of the bodies and the environment are not searched. This is the compromise we choose to obtain real time responses with accurate edge/edge collision detection.

Our CCD system ensure that the vertices and edges of the moving bodies do not get over the environment faces. It provides a robust computation of the time and location of the collisions, as well as their geometric configurations.

Many algorithms have been proposed to report collisions between moving objects. We refer the reader to existing states of the art [12], [18] for an overview. Bounding volumes hierarchies (BVH) are widely used [3], [6], [9]. Their main drawbacks concern the handling of deformations. They need additional computations to update those tree structures [2] which is usually a bottleneck for real time simulations. Moreover, such methods lead to numerous false positive tests [16].

More specialized works focus on angioscopies simulations [5], [10] where the moving bodies consist is sampled as a sets of segments. In [10], the vascular network is modeled as a tree of cells which limits domain of application. In [5] a tetrahedral subdivision is used and non-tubular structures are supported, but it imposes restrictions on the possible topological changes because of the costly updates of tetrahedrons and the lacks of efficient neighboring information. In [17] a hash function is used to map the tetrahedral subdivision and handle deformable objects. However all tetrahedrons explicitly exist which implies high memory costs that are not suitable for large simulations. Our method is similar to those develop in [5], [10], but is more general and applies to deformable environments containing inner structures.

As self-collision is not the core of the current article, we refer the reader to [7], [4], [15] for an overview of some recent optimizations in this domain.

Section II presents the model and an overview of the method. In section III an algorithm detecting edges collisions is described. It uses a particle collision detection (PCD) presented in [8]. Section IV and V present experimentation results and concludes with tracks for future works.

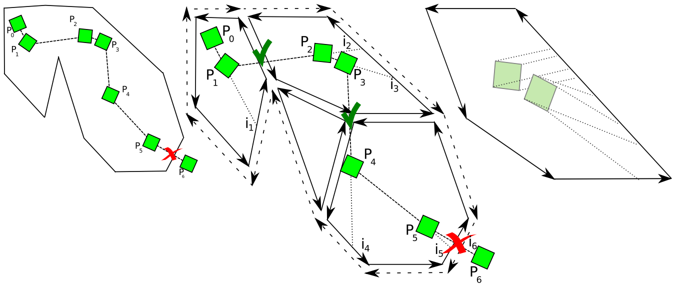

Fig. 1. Left: a moving body in a 2D scene. Middle: the decomposition of the scene and of the displacements; dotted lines represents obstacles. Right: focus on aimed at darts during the displacement.

II. METHOD OVERVIEW

The environment, the bodies are moving within, is modeled as a partition of the space in volumes, faces, edges and vertices. To optimize point/cell inclusion tests, a convexity constraint is required for every cell of this partition. Thus, 3d-cells are assumed to be convex polyhedrons with convex faces. Tetrahedral meshes are commonly used for such scene.

Our method handles any convex polyhedral decomposition and thus requires less memory for the volumetric representation.

In order to efficiently follow the vertices and to be able to handle deformations and subdivisions of its underlying mesh, the environment is structured with strong neighborhood information. We use the well known combinatorial map topological model [11]. Similar topological models have been used, in the domain of physical simulation, to treat specific operations such as gluing and cutting [13].

The aim of our system is to determine if a moving body encounters an impassable cell when moving from position P_t to position P_{t+1}. A naive approach consists in testing all possible intersections between the segment $[P_t, P_{t+1}]$ and impassable cells. For continuous detection, the faces of the extruded polyhedron obtained by sweeping the mobile from P_t to P_{t+1} should be considered.

Our method mainly consists in registering the moving objects with the space partition and to follow them along their trajectories, taking advantage of the convex partition. The displacements of the moving bodies are discretized so that CCD is possible with simple edges/faces intersection tests (figure 1). We assume that the bodies directions sparsely change and use temporal coherence to reduce the number of tests. We search and maintain the cells the object lie in, and more precisely the darts that are aimed at. Those darts define what we call *predictive tetrahedrons* for every moving body's vertices (or predictive triangles in the 2d case).

III. QUASI-CONTINUOUS COLLISION DETECTION FOR EDGES

The particle-based collision detection (PCD) system presented in [8] is a flexible approach that efficiently handles large data in meshless simulation. It reports collisions between a set of moving particles and an environment decomposed in convex polyhedrons. Whenever a particle enter an impassable cell, the PCD system returns the time and position of the collision, the kind of encountered cell (face, edge or vertex) and its surface normale (see figure 2). The system also maintains for every particle the dimension of the cell it is moving in. Thus particles sliding along faces or edges are correctly handled.

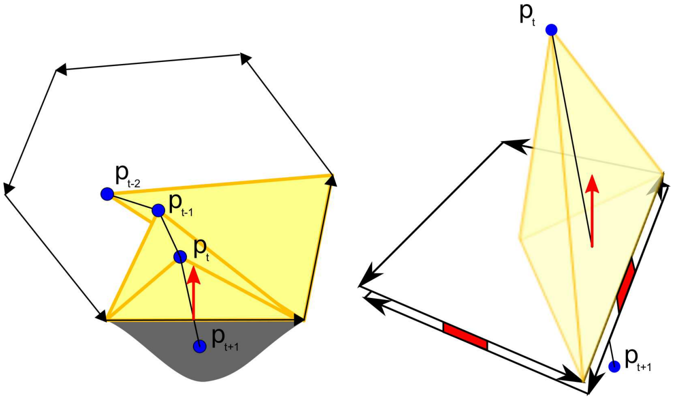

Fig. 2. Particle-based collision detection: the predictive triangles and tetrahedron are drawn in yellow. The red arrow represents the returned contact information.

During a displacement step, every vertex of the bodies is moved from position p_t to p_{t+1}. A particle is launched along $[p_t, p_{t+1}]$ using the previously mentionned PCD. Then we consider every adjacent edge between particle p_t and a neighboor q_t. Each edge sweeps over a small triangle corresponding to an elementary displacement. A collision occurs if this triangle τ intersects an impassable cell.

As the environment is modeled by a continuous space partition, an edge collision happens if and only if one of the three edges of triangle τ intersects an obstacle or if the obstacle is completely included in τ. Thanks to the small time steps used in simulations, we can assume that those elementary triangles are small enough to consider that no obstacles exist in τ. Thus we limit the tests to the first edge condition.

Further edge displacements are handled by alternative moves of the two particles. At time t, the edge $[p_t, q_t]$ is free of collision. Particle p_t moves towards p_{t+1} using one PCD step. The third edge of triangle τ – $[p_{t+1}, q_t]$ – is checked by spanning a temporary particle from p_{t+1} to q_t with a second PCD step. Then q_t moves toward q_{t+1} with a first PCD step and a second one from q_{t+1} to p_{t+1} (figure 3).

We first present the 2D case that best supports plane figures. The 3D case is detailled thereafter.

Whenever an edge collision is reported, the PCD step returns the first impasssable cell encountered along the edge. This indicates a collision with an obstacle but does not inform on the impact point (figure 4). If precise contact information is needed, two methods can be used.

A dichotomic search may be used, iterating along the $[p_t, p_{t+1}]$ edge, until the nearest segment with no contact

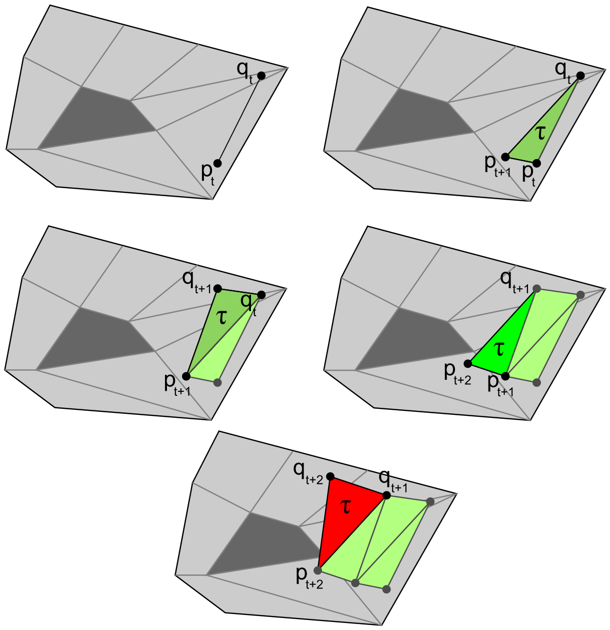

Fig. 3. Successive edges displacements and the elementary sweeping used to detect collisions.

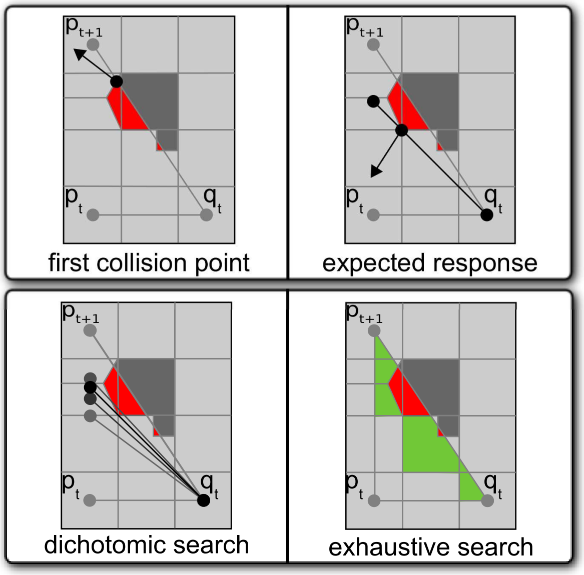

Fig. 4. Retrieving contact position and normal.

is found (figure 4 bottom left). It reports approximated, but numerically reliable, contact information with accurate time of collision. The normal is computed at the vertex of the last intersected dart.

When exact contact information is needed, an exhaustive search can be performed. It consists in searching all obstacles vertices of all the cells that are traversed by the PCD iteration. The vertex with the smallest angle is reported as the contact point and used to compute the normal. The search can be limited to the vertices that lie below segment $[p_{t+1}, q_t]$ (in green figure 4 bottom right).

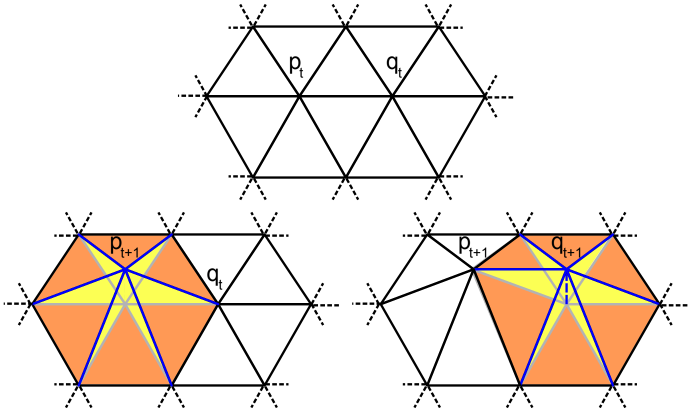

Fig. 5. Successive edges displacement for a 3D mesh.

The same approach is used in the 3D case: see figure 5. If in the 2D case all collisions are detected assuming that no small obstacles exist. In dimension 3, things are more complicated and only edges/environment collisions are reported.

When the obstacles present sharp features (represented as blue polyhedrons in figure 6), the corresponding vertices may interpenetrate the moving bodies through faces.

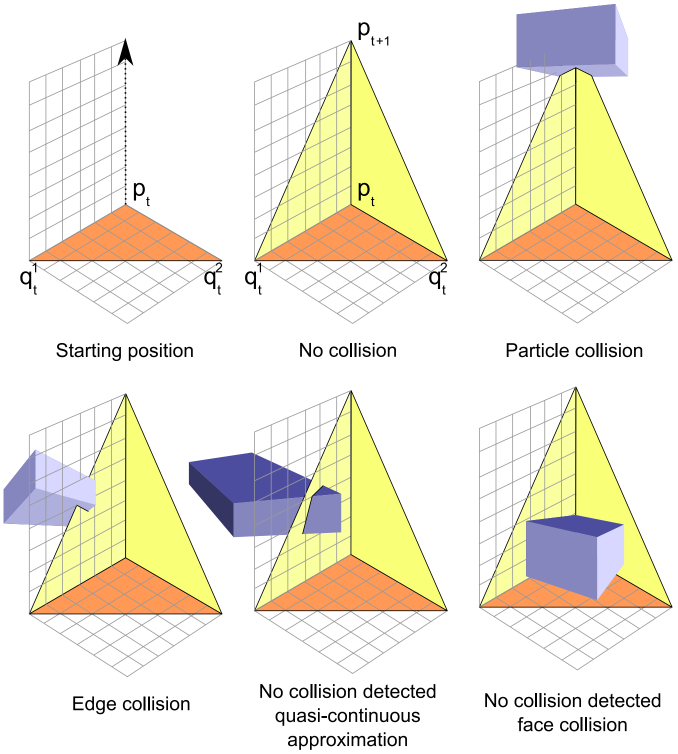

Fig. 6. Edges collisions cases and missed faces collisions

The largest penetration possible corresponds to an inverted pyramid whose edges reposed on the moving mesh edges. The volume of this pyramid is bounded by the face of the moving bodies and by the displacement length of a single vertex.

IV. EXPERIMENTATION

The proposed simulation has been implemented on a PC with a 2.4GHz Intel Core2 CPU and 2GB memory and executed on one core.

We experiment our collision detection system, in a simulation where 225 deformable hexahedrons, sampled with 8 particles each, are bouncing in a simple environment. The simulation data are recorded during 1000 time steps and summed. Thus 8*225*1000 or 1 800 000 vertex moves are performed for every simulation.

The size of the simulated environment is increased for each measurements. Here, the environment is composed of n tied hexahedrons with some obstacles, we modify the number of hexahedrons to increase the complexity of the simulation, as shown in figure 7.

We use the shape matching method introduced in [14] to simulate deformable moving solids. The benchmark consists in comparing our algorithm with an AABB boundary volumes hierarchy. The implementation of the AABB BVH is taken from the Bullet physic library.

Two tests have been performed with the BVH. The first uses segment/triangles intersection tests similar to the one presented without the use of a forecast mechanism. The second uses

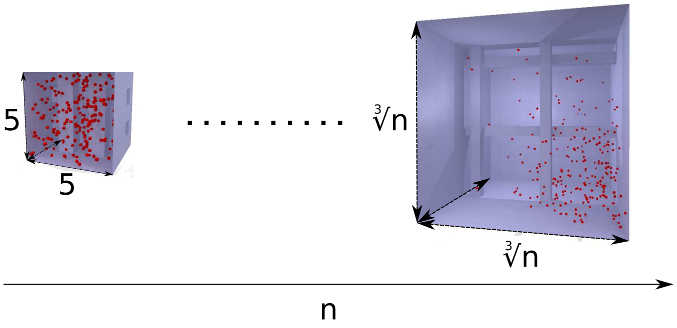

Fig. 7. Building environments of size n for the benchmark.

triangle/triangle tests, the first triangle being τ and the second a facet of the environment. The Bullet library use GIMPACT for triangle/triangle tests.

Figure 8 shows the numerical results. Our method overcome the others by a factor of 3. The segment/triangles intersection tests shows a clear logarithmic increasing with the complexity of the scene due to the underlying AABB BVH structure it use. The triangle/triangle test and our method are kept constant independently of the complexity. The triangle/triangle tests cost is amortized due to the objects repartition. Our method cost is kept constant because of the forecast mechanism used.

V. CONCLUSION

In this paper we presented an extension of a particle-based CCD for edge quasi-CCD. We plan to further extend this example to simulate endoscopies or the move of a catheter (sampled as a discretized broken line) inside the body. We plan to insert this work in a general framework dedicated to surgery simulations [1].

Not detecting collision with facets is a known limitation but the edge quasi-CCD gives already acceptable enough performance for interactive and precise simulation.

We plan to develop edges CCD by exploiting the volumetric subdivision structure in order to conserve the geometrical and topological deformations updating properties.

REFERENCES

[1] Allard, J., Cotin, S., Faure, F., Bensoussan, P.J., Poyer, F., Duriez, C., Delingette, H., Grisoni, L.: Sofa an open source framework for medical simulation. In: Medicine Meets Virtual Reality (MMVR'15), pp. 1–6 (2007)

[2] van den Bergen, G.: Efficient collision detection of complex deformable models using aabb trees. J. Graph. Tools **2**(4), 1–13 (1997)

[3] Cohen, J.D., Lin, M.C., Manocha, D., Ponamgi, M.: I-collide: an interactive and exact collision detection system for large-scale environments. In: Proc. of the symp. on Interactive 3D graphics, pp. 189–ff. (1995)

[4] Curtis, S., Tamstorf, R., Manocha, D.: Fast collision detection for deformable models using representative-triangles. In: Proc. of the symp. on Interactive 3D graphics and games, pp. 61–69. ACM (2008)

[5] Geiger, B.: Real-time collision detection and response for complex environments. In: Proc. of the Int. Conf. on Comput. Graph., pp. 105–113 (2000)

[6] Gottschalk, S., Lin, M.C., Manocha, D.: Obbtree: a hierarchical structure for rapid interference detection. In: Proc. of the 23rd annual conf. on Comput. graph. and interactive techniques, pp. 171–180 (1996)

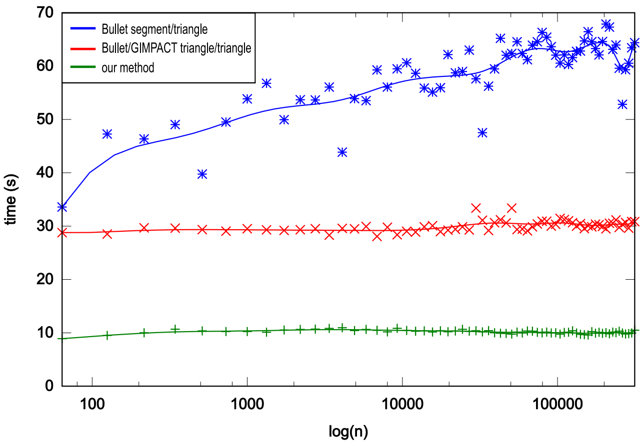

Fig. 8. Computation time for the CD in function of the complexity n of the scene for 225 deformable hexahedrons during 1000 time steps ; our method has a query time of about 10ms per time step meaning a time of approximately $5\mu s$ per particle.

[7] Govindaraju, N.K., Knott, D., Jain, N., Kabul, I., Tamstorf, R., Gayle, R., Lin, M.C., Manocha, D.: Interactive collision detection between deformable models using chromatic decomposition. ACM Trans. Graph. **24**(3), 991–999 (2005)

[8] Jund, T., Cazier, D., Dufourd, J.F.: Particle-based forecast mechanism for continuous collision detection in deformable environments. In: SPM: SIAM/ACM Joint Conf. on Geometric and Physical Modeling, pp. 147–158 (2009)

[9] Klosowski, J.T., Held, M., Mitchell, J.S., Sowizral, H., Zikan, K.: Efficient collision detection using bounding volume hierarchies of k-DOPs. IEEE Trans. on Visualization and Comput. Graphics **4**(1), 21–36 (1998)

[10] Lenoir, J., Cotin, S., Duriez, C., Neumann, P.F.: Interactive physically-based simulation of catheter and guidewire. Comput. and Graph. **30**(3), 416–422 (2006)

[11] Lienhardt, P.: N-dimensional generalized combinatorial maps and cellular quasi-manifolds. Int. J. Comput. Geometry Appl. **4**(3), 275–324 (1994)

[12] Lin, M.C., Manocha, D.: Handbook of Discrete and Computational Geometry, chap. 35 - Collision and proximity queries (2004)

[13] Lindblad, A., Turkiyyah, G.: A physically-based framework for real-time haptic cutting and interaction with 3d continuum models. In: Proc. of the ACM symp. on Solid and physical modeling, pp. 421–429 (2007)

[14] Müller, M., Heidelberger, B., Teschner, M., Gross, M.: Meshless deformations based on shape matching. ACM Trans. Graph. **24**(3), 471–478 (2005)

[15] Tang, M., Curtis, S., Yoon, S., Manocha, D.: Iccd: Interactive continuous collision detection between deformable models using connectivity-based culling. In: IEEE Trans. on Visualization and Comput. Graph., pp. 544–557 (2009)

[16] Tang, M., Manocha, D., Tong, R.: Fast continuous collision detection using deforming non-penetration filters. In: SIGGRAPH Symp. on Interactive 3D Graphics and Games (2010)

[17] Teschner, M., Heidelberger, B., Müller, M., Pomeranerts, D., Gross, M.: Optimized spatial hashing for collision detection of deformable objects. In: Vision, Modeling, Visualization, pp. 47–54 (2003)

[18] Teschner, M., Kimmerle, S., Zachmann, G., Heidelberger, B., Raghupathi, L., Fuhrmann, A., Cani, M.P., Faure, F., Magnetat-Thalmann, N., Strasser, W.: Collision detection for deformable objects. In: Eurographics State-of-the-Art Report, pp. 119–139 (2004)