A Review on Kinetic Data Structures

Simena Dinas and José María Bañón

Universidad del Valle, Facultad de Ingeniera,

Cali, Colombia

{simena.dinas,jose.banon}@correounivalle.edu.co

http://www.univalle.edu.co

0000-0003-3771-152X

Fecha Recepción: 02 DIC 2020

Fecha Aceptación: 04 DIC2020

1 Abstract

Kinetic Data Structures are a special way to model continuous collision detection due to the model and combination of the state of geometrical and motion configuration [Gui98], [GKR04], [Gui04] and [Got08]. Kinetication is a process focused on transforming an algorithm working on static data into a data structure valid for continuously changing data. A Kinetic Data Structure is a structure using some attributes in the input set, it includes a set of proofs that guarantee the attributes stay valid, and has a system that updates the data structure when the proofs or the structure of the attributes are no longer valid. The most widely utilized Kinetic Data Structures are: Collision Detection, Delaunay Triangulation, Convex Hull, Sorted List, Spanner, Voronoi, amongst others.

2 Kinetic Data Structures

A Kinetic Data Structure uses some attributes in the input set, includes a set of proofs that guarantee the attributes stay valid and, has a system that updates the data structure and when the proofs are not valid. Some concepts are important to Kinetic Data Structures:

- Data Structure: For a Kinetic Data Structure over an attribute A, there are geometric relations that certify the combinatorial structure of A, the set of rules for repairing A and the relation between certificates when a relation fails.
- Certificate: A certificate is one basic geometric relation.
- Motion Plan: An explicit description of the motion of an object in an specific time.
- Event: An event is the failure of a certificate. Events are divided into external and internal. An external event happens when the combinatorial structure of A changes, whereas an internal event happens when the certification needs to change (the structure does not change).
- Event queue: According to their earliest failure time, all certificates are placed in an event queue.

A set of criteria is defined to evaluate the Kinetic Data Structures [BGSZ97], [KSS00], [Gui01] and [Aba07].

- Responsiveness: Is related to the time required to update a certificate failure, which should be small. Accordingly, the configuration function changes, and it is necessary to find the new value and to repair the affected certificates. This proof includes identification of the failure, deletion of the bad certificates, insertion of new certificates and changes in the event queue.
- Efficiency: The efficiency of a Kinetic Data Structure is related to the number of events processed in the worst case, in other words, the quantity of events that are processed when a certificate failure happens.
- Compactness: A Kinetic Data Structure is compact when the number of certificates involved in a proof is small; each proof involves a set of certificates, a kinetic data

structure compact involves a minor number of certificates, therefore, it is reflected in the size of the event queue. On the other hand, each certificate involves a set of objects; however, each object cannot be involved in each certificate. The size of a KDS was defined as the maximum number of certificates [Gui98].

- Locality: A KDS is considered local when the number times of an object appear in certificates is small or the maximum number of certificates involving an object. It should be small. Then, the locality of a KDS is the maximum number of certificates in which any one object can ever appear.

Kinetic Data Structures were early documented focused on Collision Detection [BGSZ97] and [BGH97]. However, an in-depth study was proposed by Guibas et al. (2004) and Guibas (2004). In contrast, Basch (1999) defined the correctness criteria for Kinetic Data Structures: Responsiveness, Efficiency, Compactness and Locality. That criterion can be used to measure and compare the performance of the Kinetic Data Structure. Kinetic Data Structures have been used to model collision detection problems, usually called Kinetic Collision Detection. Kinetic collision detection between two simple polygons was proposed [BEG+04]. The authors constructed a triangulation using the inner spaces between a pair of polygons called External Relative Geodesic Triangulation - ERGT by using a subdivision of the planar space. However, the triangulation is created using the free-space between non-convex polygons and vertices. Consequently, a pseudo-triangulation is created and modified according to the response caused by colliding.

3 Kinetic Data Structure for Moving Objects

The following are the most common Kinetic Data Structures, all of them are focused on keeping track of moving objects during the time; however, the objects are different for each one:

- Delaunay Triangulation: Delaunay Triangulation is the most famous computational geometry; it takes advantage of the connection between points because they are the closest points. Several works on Kinetic Delaunay Triangulations [Rus07], [DM08], [MD08], [VK08] and [AGG+10].
- Voronoi: A Voronoi is a dual of Delaunay Triangulation; it is a special way of space decomposition [GD06].
- Regular Triangulation: A regular triangulation, usually called aWeighted Delaunay Triangulation, is similar to Delaunay triangulations, but each vertex has a real-valued weight, which depends on the applications [BSDMH05] and [Rus07].
- Delaunay Stable Graphs: It is a graph based on Delaunay Triangulation; it retains some properties of the Delaunay Triangulation to decrease the time of update by means of a dynamic subgraph [AGG+10].
- Convex Hull: The convex hull is the convex envelope of a set of points, for moving points; several authors have worked on maintaining the structure [Bas99], [ABT07], [ABTT08], [KRS10] and [dRS11].
- Collision Detection: It is popular structures based on triangulations, but they extend to non-convex polyhedral by using pseudo-triangulations [Spe01]; [GXZ01]; [AdhPS06b]; [AdhPS06a] and [BEG+04].
- Sorted Order: It is usually called Kinetic Sorted problem, it is based on maintaining organized a data structure on a set of real numbers during the motion [Ad05] and [AAdY06].
- Closest Pair: It is focused on calculating the minimum distance

between all pairs in a set, as a result the identity of the closest pair [Bas99] and [AKS08].

- All Nearest Neighbors: For every point p in a set of points, to calculate the nearest neighbor point by means of Euclidean distance [AKS08] and [SSV07].
- Separation List: In order to split the time into intervals for the continuous collision detection problem, a separation list of a set of points, and determine exactly the time of contact. Thus, Kinetic Separation List takes advantage of the temporal and spatial coherence to check between bounding volumes [WZ06].
- Minimum Spanning Circle: For a set of points P the minimum spanning circle is a circle containing all points of P [DEGS10] and [AEGH98].
- Minimum Spanning Trees: For a set of points P and a distance function, the spanning tree is a weighted complete graph on P where the nodes are points in P and the weight of each edge is the distance between the endpoints of the edge [ZY00] and [AEGH98].
- Spanners: A spanner is a graph consisting of a set of vertices and a set of edges. Every edge has a weight, which is related to the Euclidean distance between endpoints. Every vertex is associated to an edge; consequently, there is a path for each pair of vertices. The weight of a path is the sum of the weight of each edge contained in the path [AdG08] and [Ad09].
- Mesh Refinement: It is a technique based on representing three-dimensional objects with complex shapes in which most important operations are tessellations (usually triangulations) and displacements [AHT11].
- Sweep and Prune: Sweep and Prune methods are useful for collision detection; they limit the number of potential collisions on broad phase; nevertheless, the kinetic approach is based on maintaining a sorted list of one-dimension moving points during the time by using a Kinetic Sorted List [CS05].
- Tournament Tree: A tournament tree of n items is a balanced binary tree with n leaves, where interior nodes are filled from bottom up. It is a divide-and-conquer approach to organizing a set of numbers [AAdY06].
- Range Tree and KD-Trees: Both are Geometric Data Structures; they are based on Kinetic Sorted Lists. [BGH97]; [AAdY06].
- Others: Kinetic Red-Blue Minimum Separating Circle [CDZ11]; Kinetic Robust K-Center [FM10]; Soft Kinetic Data Structures [CS01]; Kinetic Bounding Volume Hierarchies [ZW06]; Kinetic Planar Subdivisions [ABd+00]; Kinetic Binary Space Partitions [AEG98]; Kinetic Graphs [HBF07]; Relative Convex Hull and Minimal Link Separation [Her04]; among others.

4 Applications

Several Kinetic Data Structures can be found in the literature; however, some are widely used tor continuous collision detection. For instance, kinetic collision detection with a fast flight plan changes caused by the collision response in a translational movement of polygons was proposed [Her04]. It used a Relative Convex Hull, which is the shortest cycle of a polygon that separates from other polygons and Minimal-Link Separation, which is a simple polygon with as fewer edges as possible that separates a polygon from other polygons. This approach was focused on reducing the time required to update the collision response.

Minimal-Link Separation was the main research in a study of kinetic data structures in collision detection [KSS00] and [KS01]; it was based on multiple non-convex planar objects. The data structures included Convex Chains to represent the objects and Pseudo-Triangulations to subdivide the free space. Furthermore, in this work a Kinetic Separation Structure (KSS) based on pseudo-triangulations was introduced. Additionally, a corridor structure to represent free space in the pseudo-triangulation was also introduced.

Nevertheless, an important state-of-the-art data structure was studied called Updating Delaunay Triangulations for moving points [DM08], it focused on a discrete, almost-Delaunay approach and kinetic Delaunay Triangulation. Comparatively, can be reviewed an analysis of relevant literature on Kinetic Data Structures [Got08]. Both documents are important to find relevant publications on Delaunay Triangulation. However, the former is focused on Delaunay Triangulation and introduced the most important approaches, whereas the latter gives a general background on Kinetic Data Structures including Delaunay Triangulations.Several works are focused on small perturbations; they exploit temporal and spatial coherence, in other words, they work on the stability or predictability of the data. For instance, researches on Delaunay Triangulation of moving points were addressed by [Vom08a], [Vom08b], [MD08] and [MTAD09]. Vomácka proposed a numerical-analytical method to compute the times of topological events required to maintain a Delaunay Triangulation. The research conducted by Machado Manhes de Castro & Devillers (2008) was focused on showing an efficient way to update Delaunay Triangulations when there are small perturbations on the vertices. In addition, a study of the times of the topological events using Delaunay triangulation was developed [VK08]. They worked on the computation of a highly time consuming process related to analyzing the topological events required to maintain a Kinetic Delaunay Triangulation with points moving on linear trajectories. In addition, fast updating of Delaunay Triangulation of moving points by bi-cell filtering was proposed by Zhou et al. (2010). The problem addressed by the authors was the slight perturbation on data for Delaunay Triangulation. They used a Bi-cell filtering to take advantage of the connectivity between the points to updating the triangulation.

To maintain a Delaunay triangulation of moving points in a distribute environment was focused on Massively Multiplayer On-Line Games (MMOG) [YCLL05]. Each node (processor) is represented as an object (player), then the objects motion is related to the kinetication of Delaunay Triangulation; however, Delaunay is updated locally for each processor without using a centralized processor or global information. Delaunay Triangulation is used to get information on proximity of objects; it reduces the number of proximity queries to adjacent points depicted by edges.

A background on kinetic data structure and applications was widely studied [Rus07], whereas a package for exact computation was developed by [RKG07]. In general terms, the work of Russel (2007) and Russel et al. (2007) was focused on the construction of a framework for Kinetic Data Structure for CGAL Computational Geometry Algorithms Library); nevertheless, the validation of the framework is based on a Kinetic Delaunay Triangulation. An analysis of Kinetic Data Structures with spherical shape and free-flying polyhedral was developed by Abam et al. (2006) and Abam (2007). The authors were based on the construction of different size spherical polyhedral; the goal was focused on reducing the number of certificates from quadratic to near-linear.

This work was conducted for similarly sized and arbitrarily sized fat objects or ball rolling. In contrast, a Kinetic Data Structure was developed for collision detection between polygons by using triangulation of free space and corridors, among others [KSS00], whereas a Kinetic Data Structure based on Bounding Volume Hierarchy was developed by Zachmann & Weller (2006); it was called a Kinetic Box Tree. The Kinetic Box Tree was based on Aligned-Axis-Bounding- Box (AABB) [ZW06]. Binary Space Partitioning (BSP) and Steiner triangulation are combined by Agarwal et al. (2000) to produce a Lower Bound Kinetic Data Structure.

An interesting geometric structure used in Kinetic Data Structure is the Convex Hull, which is the boundary of a Delaunay Triangulation. Some works on Convex hull include an approach to calculate the expansion of gas molecules by means of a self-adjusting technique and a robust implementation on three dimensions; it was reported by [ABT07] and [ABTT08], respectively. The former is based on a self-adjusting technique to identify the convex hull of gas molecules whereas, the latter proposed a robust Kinetic Convex Hull based on three stages i) a self-adjusting computation library, ii) an incremental 3D convexhull algorithm, and iii) the motion simulator. A mathematical study of Delaunay Triangulation and Convex Hull applied to kinetic data structures was proposed by using randomized trajectories set [KRS10]; it was implemented by means of pseudo-triangulations and combined dynamic and kinetic changes in order to decrease the number of operations.

Kinetic Stable Delaunay Graphs were defined by [AGG+10], to reduce the construction time of Kinetic Delaunay Triangulation. It is a dynamic subgraph of a Delaunay Triangulation. It retains many properties of a full Delaunay Triangulation and reduces upper bound on the number of topological changes.

Sorted Lists are used as Kinetic Data Structures to finding an organized list of numbers that represent a line in the space, in this sense; Kinetic Sorted List was used to maintain 1D lists for moving-sorted points [CS05], [CS06]. The Sorted List problem can be reduced to Delaunay Triangulation, Voronoi Diagrams and Convex Hull problems; however, the choice depends on the application [Ad05]. Spanners are data structures widely explored on Kinetic Data Structure. A Spanner is a graph constructed using a set of points and a set of undirected weighted straight edges connecting pair of points. It can be used to model real environments as road networks, and telecommunication networks amongst others [AdG08], [Ad09]. In contrast, Kinetic Mesh Refinement tries to maintain a mesh of continuous moving points based on the connectivity and deals with slight changes in the data; [AHT11] and [ZSW+10].

References

[AAdY06] Mohammad Ali Abam, Pankaj K. Agarwal, Mark de Berg, and Hai Yu. Out-of-order event processing in kinetic data structures. In ESA'06, pages 624-635, 2006.

[Aba07] Mohammad Ali Abam. New Data Structures and Algorithms for Mobile Data. PhD thesis, Technische Universiteit Eindhoven, 2007.

[ABd+00] Pankaj K. Agarwal, Julien Basch, Mark de Berg, Leonidas J. Guibas, and John Hershberger. Lower bounds for kinetic planar subdivisions. Discrete & Computational Geometry, pages 721-733, 2000.

[ABT07] Umut A. Acar, Guy E. Blelloch, and Kanat Tangwongsan. Kinetic 3d convex hulls via self-adjusting computation. In Jeff Erickson, editor, Symposium on Computational Geometry, pages 129-130. ACM, 2007.

[ABTT08] Umut A. Acar, Guy E. Blelloch, Kanat Tangwongsan, and Duru Türkoglu. Robust kinetic convex hulls in 3d. In Proceedings of the 16th annual European symposium on Algorithms, ESA '08, pages 29-40, Berlin, Heidelberg, 2008. Springer-Verlag.

[Ad05] Mohammad Ali Abam and Mark de Berg. Kinetic sorting and kinetic convex hulls. In Joseph S. B. Mitchell and G□unter Rote, editors, Symposium on Computational Geometry, pages 190-197. ACM, 2005.

[Ad09] Mohammad Ali Abam and Mark de Berg. Kinetic spanners in rd. In John Hershberger and Efi Fogel, editors, Symposium on Computational Geometry, pages 43-50. ACM, 2009.

[AdG08] Mohammad Ali Abam, Mark de Berg, and Joachim Gudmundsson. A simple and efficient kinetic spanner. In Monique Teillaud, editor, Symposium on Computational Geometry, pages 306-310. ACM, 2008.

[AdhPS06a] Mohammad Ali Abam, Mark de Berg, Sheung hung Poon, and Bettina Speckmann. Kinetic collision detection for balls rolling on a plane, 2006.

[AdhPS06b] Mohammad Ali Abam, Mark de Berg, Sheung hung Poon, and Bettina Speckmann. Kinetic collision detection for convex fat objects. In Proceedings of the 14th conference on Annual European Symposium - Volume 14, pages 4-15, London, UK, 2006. Springer-Verlag.

[AEG98] Pankaj K. Agarwal, Jeff Erickson, and Leonidas J. Guibas. Kinetic binary space partitions for intersecting segments and disjoint triangles. In Proceedings of the ninth annual ACM-SIAM symposium on Discrete algorithms, SODA '98, pages 107-116, Philadelphia, PA, USA, 1998. Society for Industrial and Applied Mathematics.

[AEGH98] Pankaj K. Agarwal, David Eppstein, Leonidas J. Guibas, and Monika R. Henzinger. Parametric and kinetic minimum spanning trees. In Proceedings of the 39th Annual Symposium on Foundations of Computer Science, FOCS '98, pages 596-, Washington, DC, USA, 1998. IEEE Computer Society.

[AGG+10] Pankaj K. Agarwal, Jie Gao, Leonidas J. Guibas, Haim Kaplan, Vladlen Koltun, Natan Rubin, and Micha Sharir. Kinetic stable delaunay graphs. In Proceedings of the 2010 annual symposium on Computational geometry, SoCG '10, pages 127-136, New York, NY, USA, 2010. ACM.

[AHT11] Umut A. Acar, Benoît Hudson, and Duru Türkoglu. Kinetic mesh refinement in 2d. In Proceedings of the 27th annual ACM symposium on Computational geometry, SoCG '11, pages 341-350, New York, NY, USA, 2011. ACM.

[AKS08] Pankaj K. Agarwal, Haim Kaplan, and Micha Sharir. Kinetic and dynamic data structures for closest pair and all nearest neighbors. ACM Trans. Algorithms, 5:4:1-4:37, December 2008.

[Bas99] Julien Basch. Kinetic Data Structures. PhD thesis, Stanford University, Stanford, CA, USA, 1999.

[BEG+04] Julien Basch, Jeff Erickson, Leonidas J. Guibas, John Hershberger, and Li Zhang. Kinetic collision detection between two simple polygons. Comput. Geom. Theory Appl., 27:211-235, March 2004.

[BGH97] Julien Basch, Leonidas J. Guibas, and John Hershberger. Data structures for mobile data. In SODA, pages 747-756, 1997.

[BGSZ97] Julien Basch, Leonidas J. Guibas, Craig D. Silverstein, and Li Zhang. A practical evaluation of kinetic data structures. In Proceedings of the thirteenth annual symposium on Computational geometry, SCG '97, pages 388-390, New York, NY, USA, 1997. ACM.

[BSDMH05] Tilo Beyer, Gernot Schaller, Andreas Deutsch, and Michael Meyer-Hermann. Parallel dynamic and kinetic regular triangulation in three dimensions. Computer Physics Communications, 172(2):86-108, November 2005.

[CDZ11] Yam Ki Cheung, Ovidiu Daescu, and Marko Zivanic. Kinetic red-blue minimum separating circle. In Proceedings of the 5th international conference on Combinatorial optimization and applications, COCOA'11, pages 448-463, Berlin, Heidelberg, 2011. Springer-Verlag.

[CS01] Artur Czumaj and Christian Sohler. Soft kinetic data structures. In Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms, SODA '01, pages 865-872, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[CS05] Daniel S. Coming and Oliver G. Staadt. Kinetic sweep and prune for collision detection. In In Proc. Workshop on Virtual Reality Interactions and Physical Simulations, pages 81-90, 2005.

[CS06] Daniel S. Coming and Oliver G. Staadt. Kinetic sweep and prune for multi-body continuous motion. Computers & Graphics, 30:439-449, 2006.

[DEGS10] Erik D. Demaine, Sarah Eisenstat, Leonidas J. Guibas, and André Schulz. Kinetic minimum spanning circle. In In Proceedings of the Fall Workshop on Computational Geometry, New York, USA, 2010.

[DM08] Olivier Devillers and Pedro Machado Manh~aes de Castro. State of the art:Updating delaunay triangulations for moving points. Rapport de recherché RR-6665, INRIA, 2008.

[dRS11] Mark de Berg, Marcel Roeloffzen, and Bettina Speckmann. Kinetic convex hulls and delaunay triangulations in the black-box model. In Symposium on Computational Geometry, pages 244-253, 2011.

[FM10] Sorelle A. Friedler and David M. Mount. Approximation algorithm for the kinetic robust k-center problem. Comput. Geom. Theory Appl., 43:572- 586, August 2010.

[GD06] Christopher M. Gold and Maciej Dakowicz. Kinetic voronoi/Delaunay drawing tools. In Proceedings of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering, ISVD '06, pages 76-84, Washington, DC, USA, 2006. IEEE Computer Society.

[GKR04] Leonidas J. Guibas, Menelaos Ioannis Karavelas, and Daniel Russel. A computational framework for handling motion. In Proceedings of the Sixth Workshop on Algorithm Engineering and Experiments, pages 129-141, 2004.

[Got08] Bo Gotthardt. Literature study in kinetic data structures, 2008.

[Gui98] Leonidas J. Guibas. Kinetic data structures: a state of the art report. In Proceedings of the third workshop on the algorithmic foundations of robotics on Robotics : the algorithmic perspective: the algorithmic perspective, WAFR '98, pages 191-209, Natick, MA, USA, 1998. A. K. Peters, Ltd.

[Gui01] Leonidas J. Guibas. Kinetic data structures, 2001.

[Gui04] Leonidas J. Guibas. Kinetic data structures. In Handbook of Data Structures and Applications. CRC Press, 2004.

[GXZ01] Leonidas J. Guibas, Feng Xie, and Li Zhang. Kinetic collision detection: Algorithms and experiments. In ICRA'01, pages 2903-2910, 2001.

[HBF07] Jérome Härri, Christian Bonnet, and Fethi Filali. Kinetic graphs: a framework for capturing the dynamics of mobile structures in manet. In Proceedings of the 2nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, PM2HW2N '07, pages 88-91, New York, NY, USA, 2007. ACM.

[Her04] John Hershberger. Kinetic collision detection with fast flight plan changes. Inf. Process. Lett., 92:287-291, December 2004.

[KRS10] Haim Kaplan, Natan Rubin, and Micha Sharir. A kinetic triangulation scheme for moving points in the plane. In Jack Snoeyink, Mark de Berg, Joseph S. B. Mitchell, G□unter Rote, and Monique Teillaud, editors, Symposium on Computational Geometry, pages 137-146. ACM, 2010.

[KS01] David Kirkpatrick and Bettina Speckmann. Separation sensitive kinetic separation structures for convex polygons. In Revised Papers from the Japanese Conference on Discrete and Computational Geometry, JCDCG '00, pages 222-236, London, UK, 2001. Springer-Verlag.

[KSS00] David Kirkpatrick, Jack Snoeyink, and Bettina Speckmann. Kinetic collision detection for simple polygons. In Proceedings of the sixteenth annual symposium on Computational geometry, SCG '00, pages 322-330, New York, NY, USA, 2000. ACM.

[MD08] Pedro Machado Manh~aes de Castro and Olivier Devillers. Delaunay triangulations for moving points. Research Report RR-6750, INRIA, 2008.

[MTAD09] Pedro Machado Manh~aes de Castro, Jane Tournois, Pierre Alliez, and Olivier Devillers. Filtering relocations on a delaunay triangulation. In Proceedings of the Symposium on Geometry Processing, SGP '09, pages 1465-1474, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association.

[RKG07] Daniel Russel, Menelaos Ioannis Karavelas, and Leonidas J. Guibas. A package for exact kinetic data structures and sweepline algorithms. Comput. Geom. Theory Appl., 38:111-127, September 2007.

[Rus07] Daniel Russel. Kinetic data structures in practice. PhD thesis, Stanford University, Stanford, CA, USA, 2007.

[Spe01] Bettina Speckmann. Kinetic Data Structures for Collision Detection. PhD thesis, Müunster Germany, 2001.

[SSV07] Jagan Sankaranarayanan, Hanan Samet, and Amitabh Varshney. A fast all nearest neighbor algorithm for applications involving large point-clouds. Comput. Graph., 31:157-174, April 2007.

[VK08] Tomás Vomácka and Ivana Kolingerová. Computation of topologic events in kinetic delaunay triangulation using sturm sequences of polynomials. In SIGRAD 2008, pages 57-64, Link□oping, 2008. University Electronic Press.

[Vom08a] Tomás Vomácka. Delaunay triangulation of moving points. In CESCG 2008, pages 67-74, Vienna, 2008. Vienna University of Technology.

[Vom08b] Tomás Vomácka. Delaunay Triangulation of Moving Points in the Plane. PhD thesis, University of West Bohemia, Pilsen, Czech Republic, 2008.

[WZ06] René Weller and Gabriel Zachmann. Kinetic separation lists for continuous collision detection of deformable objects. In VRIPHYS'06, pages 33-42, 2006.

[YCLL05] Taewon Yoo, Sunghee Choi, Hyonik Lee, and Jinwon Lee. Distributed kinetic delaunay triangulation, 2005.

[ZSW+10] Yuanfeng Zhou, Feng Sun, Wenping Wang, Jiaye Wang, and Caiming Zhang. Fast updating of delaunay triangulation of moving points by bi-cell filtering. Computer Graphics Forum, 29(7):2233-2242, 2010.

[ZW06] Gabriel Zachmann and Rene Weller. Kinetic bounding volume hierarchies for deformable objects. In Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications, VRCIA '06, pages 189-196, New York, NY, USA, 2006. ACM.

[ZY00] Dongliang Zhang and Matthew M. F. Yuen. Collision detection for clothed human animation. In Proceedings of the 8th Pacific Conference on Computer Graphics and Applications, PG '00, pages 328-, Washington, DC, USA, 2000. IEEE Computer Society.