A Multi-Scale Singularity Bounding Volume Hierarchy

Kerawit Somchaipeng
3D-Lab, School of Dentistry
Dept. of Pediatric Dentistry
University of Copenhagen
Nørre Alle 20, DK-2200
Copenhagen N, Denmark
kerawit@lab3d.odont.ku.dk

Kenny Erleben
Dept. of Computer Science
University of Copenhagen
Universitetsparken 1
DK-2100, Copenhagen N
Denmark
kenny@diku.dk

Jon Sporring
Dept. of Computer Science
University of Copenhagen
Universitetsparken 1
DK-2100, Copenhagen N
Denmark
sporring@diku.dk

ABSTRACT

A scale space approach is taken for building Bounding Volume Hierarchies (BVHs) for collision detection. A spherical bounding volume is generated at each node of the BVH using estimates of the mass distribution.
Traditional top-down methods approximates the surface of an object in a coarse to fine manner, by recursively increasing resolution by some factor, e.g. 2. The method presented in this article analyzes the mass distribution of a solid object using a well founded scale-space based on the Diffusion Equation: the Gaussian Scale-Space. In the Gaussian scale-space, the deep structure of extremal mass points is naturally binary, and the linking process is therefore simple.
The main contribution of this article is a novel approach for constructing BVHs using Multi-Scale Singularity Trees (MSSTs) for collision detection. The BVH-building algorithm extends the field with a new method based on volumetric shape rather than statistics of the surface geometry or geometrical constructs such as medial surfaces.

Keywords
Collision Detection, Bounding Volume Hierarchies, Gaussian Scale Space

1 INTRODUCTION

In physics-based animation, collision detection often becomes the bottleneck, since a collision query needs to be performed in every simulation step in order to determine contacting and colliding objects. Animations can have many objects, all of which may have a complex geometry, such as polygonal soups of several thousands facets, and it is therefore a computationally heavy burden to perform collision detection especially for real-time interaction.
Bounding Volume Hierarchies (BVHs) are widely used in computer graphics, e.g. for ray tracing [GS87], and they are quite popular in animation (e.g. [BMF03] uses them for cloth animation), since they are applicable of handling more general shapes than most feature-based and simplex-based algorithms, they tend to generate smaller hierarchies than spatial subdivision algorithms, and they offer a graceful degradation of objects, which is highly useful when accuracy is to be traded for performance. New performance improvements of BVHs is therefore of great practical and theoretical interest to the computer graphics and animation community.
The main contribution of this paper is a novel algorithm for bottom-up construction of spherical approximating BVHs. We prefer our hierarchies, firstly because they save memory, and therefore increases simulation performance, when compared to traditional BVH, and secondly because they are a direct implementation of the mass of objects rather than their boundary representation.
In this article we will restrain ourselves from the n-body problem and only consider narrow phase [Hub93] collision detection of solid non-deformable objects.

Conference proceedings ISBN 80-903100-7-9
WSCG'2005, January 31-February 4, 2005
Plzen, Czech Republic.

1.1 PREVIOUS WORK

There is a wealth of literature on collision detection, and many different approaches have been investigated. Spatial subdivision algorithms like Bi-

nary Space-Partitioning (BSP) tree [Mel01], octree [TC96, GDO00, ES03], k-d trees and grids [GDO00, ES03], feature-based algorithms like polygonal intersection [MW88], Lin-Can [PML97], VClip [Mir98], SWIFT [EL01], recursive search methods [SL00], simplex-based such as GJK [GJK88, vdB01], generalized Voronoi diagrams [HKL$^+$99], and signed distance maps [GBF03, BMF03, Hir02]. Finally there are algorithms based on BVHs such as ours.

BVHs have been around for a long time. Consequently there is a huge wealth of literature about BVHs. Most of the literature addresses homogeneous BVHs and top-down construction methods. A great variety of different types of bounding volumes have been reported: Spheres [Hub96, Pal95, DO00], axed aligned bounding boxes (AABBs) [Ber97, LAM01], oriented bounding boxes (OBBs) [GLM96, Got00], discrete orientation polytypes (k-DOPs) [KHM$^+$98, Zac98], Quantized Orientation Slabs with Primary Orientations (QuOSPOs) [He99], Spherical shell [KPLM98], and swept sphere volumes (SSVs) [LGLM99]. In general, it has been discovered that there is a trade-off between the complexity of the geometry of a bounding volume and the speed of its overlap test and the number of overlap tests in a query.

In contrast to bounding volumes types, there has only been written little on approximating BVHs. To our knowledge [Hub93] pioneered the field, where octrees combined with simulated annealing were used to construct a sphere tree, followed by [PG95, Pal95], cumulating with a superior bottom-up construction method based on medial surface (M-reps) [Hub96]. More recently [OD99, DO00] used approximating sphere-trees built in a top down fashion based on an octree for time critical collision detection, and [BO04] used an adaptive m-rep approximation-based top-down construction algorithm.

There have been written even less about heterogeneous bounding volume hierarchies, although object hierarchies of different primitive volume types are a widely used concept in most of todays simulators [Ode, Vor, Kar]. The SSVs [LGLM99] are one of the most recent publications. The general belief is, however, that heterogeneous bounding volumes does not change the fundamental algorithms, but merely introduces a raft of other problems. It is also believed that heterogeneous bounding volumes could provide better and more tightly fitting bounding volumes resulting in higher convergence towards the true shape volume of the objects. This could mean an increase in the pruning capabilities and a corresponding increase in performance.

Most of the work with BVHs has addressed objects that are represented by polygonal models. Many experiments also indicate that OBBs (and other rectangular volumes) provide the best convergence for polygonal models [GLM96, Got00, Zac98, LGLM99], while spherical volumes are believed to converge best towards the volume. The underlying query algorithms for penetration detection, separation distance and contact determination of BVHs have not changed much. In its basic form, these algorithms are nothing more than simple traversals.

To our knowledge, the trees based on the deep structure of Gaussian Scale-Space has not been used previously for generating BVHs in collision detection. An alternative to Gaussian scale-space is curvature scale-spaces, from which M-reps are derived. M-reps based methods are state of the art for bottom-up construction method [Hub96] and top-down construction [BO04]. For deformable objects such as cloth, bottom-up construction based on mesh topology [VM95, VMT00, BFA02] are the preferred choice. In [Ber97] a median based top-down method was proposed for building an AABB tree. [LAM01] suggested using a mesh connectivity-tree in a top-down construction method.

2 GAUSSIAN SCALE-SPACE

The $N+1$ dimensional Gaussian scale-space, $L : \mathbb{R}^{N+1} \rightarrow \mathbb{R}$, of an N dimensional image, $I : \mathbb{R}^N \rightarrow \mathbb{R}$, is an ordered stack of images, where each image is a blurred version of the former [Iij62, Wit83, Koe84]. The blurring is performed according to the diffusion equation,

$$\partial_t L = \nabla^2 L \ , \tag{1}$$

where $\partial_t L$ is the first partial-derivative of the image in the scale direction t, and ∇^2 is the Laplacian operator, which in 3 dimensions reads $\partial_x^2 + \partial_y^2 + \partial_z^2$.

An example of the scale-space of a three-dimensional solid cow is shown in Fig. 1. The continuous scale parameter enables smooth degradation of the object detail.

The Gaussian kernel is the Green's function of the heat diffusion equation, i.e.

$$L(\cdot;t) = I(\cdot) \otimes g(\cdot;t) \ , \tag{2}$$

$$g(x;t) = \frac{1}{(4\pi t)^{N/2}} e^{-x^T x/(4t)} \ , \tag{3}$$

where $L(\cdot,t)$ is the image at scale t, $I(\cdot)$ is the original image, $\otimes$ is the convolution operator, $g(\cdot;t)$ is the Gaussian kernel at scale t, N is the dimensionality of the problem, and $t = \sigma^2/2$, using σ as the standard deviation of the Gaussian kernel. The *Gaussian scale-space* is henceforth called the scale-space in this article. The information in scale-space is logarithmically degraded, the scale-parameter is therefore often sampled exponentially using $\sigma = \sigma_0 e^T$. Since differentiation commutes with convolution and the Gaussian

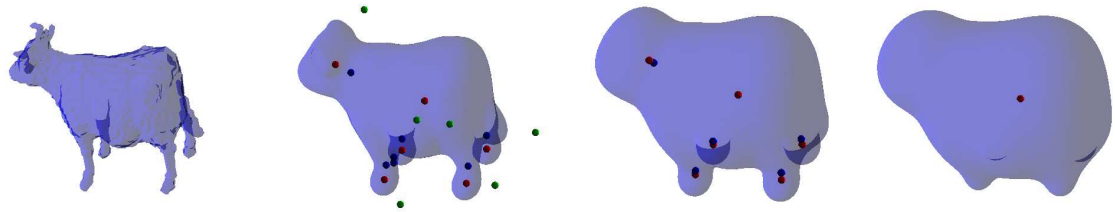

Figure 1: An example of the scale-space of a solid cow [Bra]. From left to right, the images show the zero iso-surfaces of the solid cow at scales $\sigma = 0, 2, 3.75, 5$. The small red, green, and blue spheres denote maxima, minima, and saddles, respectively

kernel is infinitely differentiable, differentiation of images in scale-spaces is conveniently computed,

$$\partial_{x^n} L(\cdot;t) = \partial_{x^n} \left(I(\cdot) \otimes g(\cdot;t) \right) = I(\cdot) \otimes \partial_{x^n} g(\cdot;t) \ . \tag{4}$$

Alternative implementations of the scale-space are multiplication in the Fourier Domain, finite differencing schemes for solving the heat diffusion equation, additive operator splitting, and recursive implementation [Der92]. We prefer the spatial convolution, since it is guaranteed not to introduce new extrema in homogeneous regions. Typical border conditions are Dirichlet, Cyclic repetition, and Neumann boundaries. We use Dirichlet boundaries, where the image is extended with zero values in all directions.

Although the dimensionality of the constructed scale-space is one higher than the dimensionality of the original image, *critical points*, in the image at each scale are always points. A critical point is e.g. an extremum, $\partial_x L = \partial_y L = \partial_z L = 0$. Critical points are classified by the eigenvalues of the Hessian matrix, the matrix of all second derivatives, computed at that point. As we increase the scale parameter, the critical points move smoothly forming *critical paths*. Along scale, critical points meet and annihilate or are created. Such events are called catastrophe events, and the points where they occur are called catastrophe points. The collection of events is called the *deep structure* of the image. The notion of genericity is used to disregard events that are not likely to occur for typical images, i.e. generic events are stable under slight perturbation of the image. There are only two types of generic catastrophe events in scale-space namely pairwise *creation events* and *annihilation events* [Dam97], and it has further been shown that generic catastrophe events only involves pairs of critical points where one and only one eigenvalue of the Hessian matrix changes its sign, e.g. the annihilation of a minimum (+, +, +) and a saddle (+, +, -). The implementation detail of the method for extracting critical paths and catastrophe points in 3+1D scale-space can be found in [SSKJ03].

3 MSSTs

Multi-Scale Singularity Trees (MSSTs) are scale-space based multi-scale image representation. They are constructed based on the nesting of image features in the scale-space to represent the deep structure of the original image. Two kinds of MSSTs are introduced in [SSKJ05]: Extrema-Based MSSTs and Saddle-Based MSSTs. Extrema-Based MSSTs will be discussed in this article. The method produces rooted ordered binary trees with catastrophe points as nodes. In 3+1D scale-space, catastrophes are also possibly caused by creations or annihilation of saddle points, e.g. between critical points with eigenvalues of the Hessian matrix (+, +, -) and (+, -, -). These saddle-saddle annihilation catastrophes together with all creation catastrophes are ignored.

Other scale-space based methods that produce tree structure but only for up to 2+1D scale-space can be found in [LP90, Kui02].

3.1 Extrema Partitions

Given an image an any scale, we would like to partition the image at one scale into segments so that each segment contains only and exactly one extremum. Let $\Omega \subset \mathbb{R}^N$ be a compact connected domain and define $I : \Omega \to \mathbb{R}^+$ to be an image, $\vec{e} \in \Omega$ as an extremum, and $\vec{x} \in \Omega$ as an image point in the domain. Consider a set of continuous functions $\gamma : [0, P] \to \Omega$ for which $\gamma(0) = \vec{e}$ and $\gamma(P) = \vec{x}$, $\gamma \in \Gamma_{\vec{e}\vec{x}}$, where $\Gamma_{\vec{e}\vec{x}}$ is the set of all paths in the domain from the extremum $\vec{e}$ to the point $\vec{x}$, and γ is parameterized using Euclidean arc-length. We define the energy $E_{\vec{e}}(\vec{x})$ with respect to an extremum $\vec{e}$ evaluated at $\vec{x}$ as,

$$E_{\vec{e}}(x) = \inf_{\gamma \in \Gamma_{\vec{e}\vec{x}}} \int_0^P \sqrt{(\alpha - 1) \mid \frac{\partial \gamma(p)}{\partial p} \mid^2 + \alpha \mid \frac{\partial I(\gamma(p))}{\partial p} \mid^2} \, dp \ . \tag{5}$$

Note that the energy functional is independent of parameterization. When $\alpha = 1$, the energy functional is

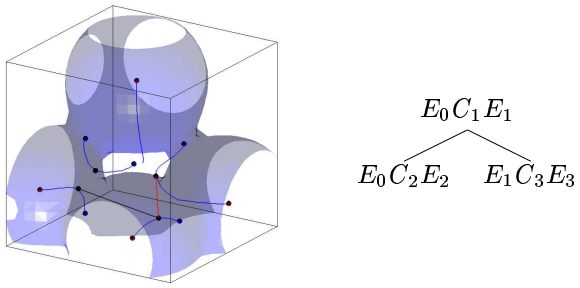

Figure 2: The Extrema-Based MSST of a three-dimensional image of four Gaussian blobs. The 2.0 iso-surfaces of the image at scale $\sigma = 3$ is shown in blue, on the left panel. The small red and blue spheres are the maxima and saddles respectively. The blue lines are the critical paths (the scale axis is projected away) and the small black spheres are the catastrophe points. The black line and the red line denote the left-child linking and the right-child linking in the tree. A schematic drawing of the extracted MSST is shown on the right panel. Note that there is a saddle-saddle catastrophe which is ignored

also known as the path variation, a generalization of the total variation [AC03]. The path variation depends solely on the image intensity and is invariant to affine transformation of the underlying space. Moreover, it is co-variant with scaling of the image intensity. If $\alpha \to 0$, the energy functional will increasingly depend on the spatial distance, and therefore become increasingly localized in space.
Let $\mathcal{E} \subset \Omega$ be the set of all extrema in the image. The *extrema partition* [AC03]., Z_i, associated with an extrema $\vec{e}_i \in \mathcal{E}$ is defined as the set of all points in the domain, where the energy $E_{\vec{e}_i}(\vec{x})$ is minimal,

$$Z_i = \left\{ \vec{x} \in \Omega \middle| E_{\vec{e}_i}(\vec{x}) < E_{\vec{e}_i}(\vec{x}), \forall \vec{e}_j \in \mathcal{E}, i \neq j \right\} . \quad (6)$$

An approximation of the energy map $M_i : \Omega \to \mathbb{R}^+$, which defines the energy at every point in the image associated with an extremum $\vec{e}_i$, can be efficiently calculated using the *Fast Marching Methods* [Set99].

3.2 Constructing MSSTs

MSSTs are defined by nodes and their relations. Each MSST node consists of three components: The image segment(i) that immediately covers the area of the image segment(ii) disappearing at the catastrophe(iii). For algorithmically convenience we denote the 'surviving' image segment the *leftport*, the catastrophe for the *body*, and the disappearing image segment for the *rightport*. Because there is exactly one image segment associated with an extremum and exactly one extremum disappears at an annihilation catastrophe, then exactly one image segment also disappears.
A node $E_{left} C_{body} E_{right}$ is generated if an image segment of E_{right} disappears at the catastrophe C_{body} inside an image segment of E_{left}. The inclusion is easily determined by calculating the energy map with respect to the catastrophe C_{body}: the image segment of E_{right} is nested in side the image segment of E_{left} if the energy evaluated at E_{left} is minimal among all extrema existing at that scale.
Assuming that critical paths and catastrophe points in the scale-space are already and correctly detected, then the MSST building algorithm is as follows:

1. Set the root of the tree as $E_{last} C_{top} E_{ann}$, where E_{last} denotes the last extremum that remains in the scale-space, C_{top} denotes the highest catastrophe in scale, and E_{ann} denotes the extremum that annihilates at the catastrophe.

2. At the highest unprocessed catastrophe C_{next} in scale, calculate the energy map with respect to the catastrophe and create a node $E_{cover} C_{next} E_{ann}$, where E_{ann} is the extremum that disappears at C_{next}, and the energy evaluated at the extremum E_{cover} is minimal among all extrema existing at that scale.

3. Link the new created node as the leftchild of a node in the tree that does not have the leftchild and where E_{cover} equals its leftport, or as the rightchild of a node in the tree that does not have the rightchild and where E_{cover} equals its rightport.

4. Repeat 2., 3., and 4. until all catastrophe points are processed.

An example of the Extrema-Based MSST constructed from a simple three-dimensional image of four Gaussian blobs is shown in Fig. 2. The constructed MSST has three nodes corresponding to the three relevant catastrophes in the scale-space. An annihilation catastrophe of a pair of saddles is ignored.

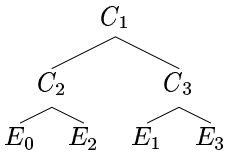

Figure 3: An extended Extrema-Based MSST example

4 BVHs from MSSTs

Given a MSST, we produce a BVH as follows. The MSST is extended with a set of leaves according to the leftport and the rightport representing each extremum in the original image. The newly added leaves represent the finest scale for the BVH. All free ports are extended with a leaf for the corresponding extremum, and then all ports are removed. The result is that the MSST is extended with one leaf for each extremum. All the extrema will appear in the extended MSST one and only one time. The extended MSST of the image in Fig. 2 is shown in Fig. 3.

We can denote the catastrophe scale as a size measure of the corresponding extremum. That is, at the catastrophe scale, the corresponding Gaussian will have a size that dominates the underlying image structures. We may also give a statistical interpretations using Tchebycheff's inequality [BM93]. It states that for a random variable X with standard deviation σ, the probability of finding values outsize a bound proportional to its standard deviation is inversely small:

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2} \tag{7}$$

We take this as a guide to set the size of the leaf bounding volumes, i.e. a leaf will be given a sphere, who's radius is proportional to the catastrophe scale. There will be one extremum, which does not disappear in a catastrophe, which is the last extremum in the scale-space. We set the bounding volume of the final extremum to be proportional to the distance to its only sibling in the MSST minus the already known sibling's radius in the BVH.

Since the BVH is binary, we find bounding volume for the non-leaf nodes in the tree as the smallest sphere that encloses the two child spheres. Although tighter bounds may be found, this is left for further development.

5 RESULTS

Currently, our algorithm is capable of producing trees from objects that are sampled on a 256^3 grid, for a reasonable computation time, we only use 64^3 grids. We demonstrate our algorithm on the cow polygonal mesh [Bra]. Figure 4 shows a schematic drawing of the extracted BVH of a solid cow and Fig. 5 shows a

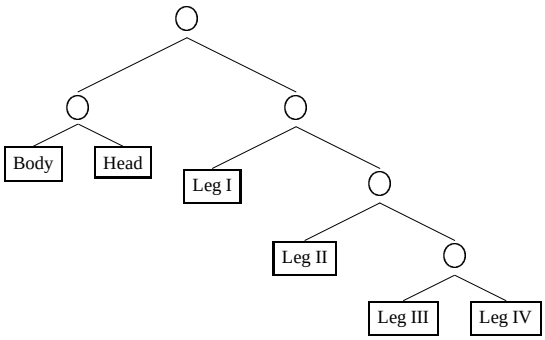

Figure 4: A schematic drawing of the extracted BVH of a solid cow

solid cow together with the spherical bounding volume at each level in the hierarchy.

In the scale-space of the cow, the legs of the cow appears in a sequential manner from coarse to fine. This makes the tree building process simple, however, in this particular example, it would possibly be more natural to let the leg-nodes appear at the same time in a 4-ary tree node. In our tree, such decisions can be enforced by post-processing, and a useful indication in this case would be that the catastrophes occur within a very narrow scale-band.

There are many properties which are interesting when evaluating the quality of a BVH. Unfortunately some of them are contradicting each other.

- Smallest possible bounding volumes
- Smallest possible overlap between volumes at the same depth in the hierarchy
- Small sized BVH, i.e. as few nodes as possible
- Complete coverage versus sampling based coverage
- "balanced" trees

The last property is one we challenge, although it has been proved that balanced trees provide best worst case queries, a balanced tree do not represent the scale of the object. Working with time critical or approximating queries this become an important property. We suggest that the tree should be balanced with respect to the density of the object.

6 DISCUSSION

Most recent work with BVHs has focused on: Trying out new kinds of bounding volumes, figuring out better methods for fitting a bounding volume to a subset of an object's underlying geometry, finding faster and better overlap test methods, and comparing homogeneous BVHs of different bounding volume types.

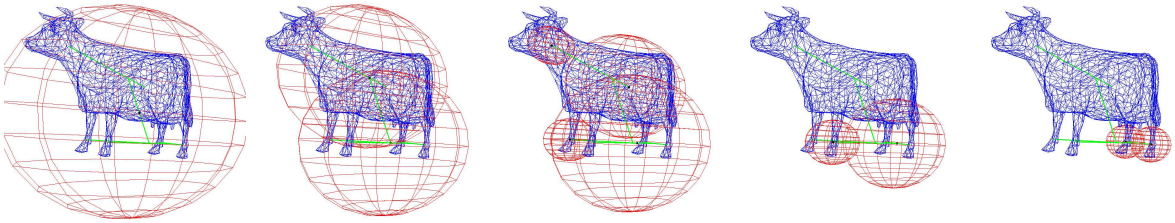

Figure 5: A solid cow and the hierarchical bounding volumes at each level of the BVH. The surface of the original cow, the links between catastrophes in the scale-space and the spherical bounding volumes are shown from left to right for the level one to five of the BVH

In order to improve the performance of traversal algorithms, depth control, layered bounding volumes, caching bounding volumes, and shared bounding volumes have been studied. We have chosen to classify our method as being a mixed bottom-up and top-down method, because the scale-space is built bottom-up, and the MSST are found in a top-down manner. The corresponding BVH is then built in a straightforward incremental way, by doing an order traversal of the MSST, and creating bounding volume nodes as catastrophes are encountered.

The computational complexity for our algorithm is currently high. Using N^3 as the number of pixels in the image, S as the number of scales to be evaluated, σ as the largest scale, K as the number of critical line-pieces found, and E as the number of extrema at the lowest scale, the computational complexity for each part of our algorithm is as follows:

Computation of the Scale-Space: $\mathcal{O}(S\sigma^3 N^3)$
: It may be possible to improve the calculation time for the Gaussian scale-space, e.g. using sub-sampled image for approximating scaled image at high scales or using faster alternatives to spatial convolution. However, we have not yet found alternatives that does not introduces spurious extrema in homogeneous regions.

Storage of the Scale-Space: $\mathcal{O}(2N^3)$
: The most memory intensive part of our algorithm is the storage of the scale-space. We only require the storage of two neighboring scales in order to find the critical paths in our current implementation.

Extracting Critical Paths: $\mathcal{O}(SN^3 + K^2)$
: The critical paths can be extracted considerably faster by tracking each extremum from the finest scale, however this would require either to store the full Scale-Space or perform local calculations during the tracking process. Since this is by far not the slowest part of our algorithm, we have left this for further research.

Finding a Euclidean Tree, $\alpha = 0$ **in (5):** $\mathcal{O}(E^2)$
: It is fastest to use the Euclidean metric in (5), for $0 < \alpha \leq 1$ see below.

Finding a General Tree: $\mathcal{O}(EN^3 \log N^3)$
: This is the most computationally expensive part of our algorithm. However, we expect that the speed of the Fast Marching Method can be improved by a narrow band implementation.

Gaussian scale space provides us with a continuous degradation of an object, other algorithms fail completely on this point, they typical control their scale by saying that at the next level of the BVH should have 50% less number of volumes, or at the next level the volumes should fit 20% better. A direct study of scales seems to be a more proper representation.

Medial surface (M-reps) based methods for building BVHs have been the approach to use for bottom-up construction. Our method differs from M-reps significantly by being a density based method, whereas M-reps is more a surface-based method. Furthermore our method provides us with a natural scale that is easily used to determine both bounding volumes and the topology of the hierarchy. M-reps do not provide this scale information nor can they tell one about the density of an object.

The well-established foundation on scale-space theory provides us with a well-defined concept of scale, shape, and detail of an object. These concepts are valuable tools as our work hopefully demonstrates.

The main contribution of our work is a new method for building bounding volume hierarchy, however, there is still much need to be done. So far our work has been a feasibility study showing that the construction of BVHs from MSSTs actually can be done. We have not yet made any attempt towards comparing the quality of the multi-scale singularity BVH with other algorithms. Future research will be on the tightening of the bounding volumes utilizing information in scale-space.

7 ACKNOWLEDGMENTS

This work is part of the DSSCV project sponsored by the IST Programme of the European Union (IST-2001-35443).

References

[AC03] P. A. Arbelaez and L. D. Cohen. The Extrema Edges". In *Scale Space 2003, LNCS 2695*, pages 180–195, 2003.

[Ber97] G. van den Bergen. Effi cient collision detection of complex deformable models using AABB trees. *Journal of Graphics Tools*, 2(4):1–13, 1997.

[BFA02] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. *Proceedings of ACM SIGGRAPH*, 21(3):594–603, 2002.

[BM93] L. Brøndum and J. D. Monrad. *Statistik I - Sandsynlighedsregning og statistiske grundbegreber*. Den private ingeniørfond, 1993.

[BMF03] R. Bridson, S. Marino, and R. Fedkiw. Simulation of clothing with folds and wrinkles. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 28–36. Eurographics Association, 2003.

[BO04] Gareth Bradshaw and Carol O'Sullivan. Adaptive medial-axis approximation for sphere-tree construction. *ACM Transactions on Graphics*, 23(1), January 2004.

[Bra] Gareth Bradshaw. Polygonal mesh of a cow. .

[Dam97] James Damon. Local Morse theory for Gaussian blurred functions. In Jon Sporring, Mads Nielsen, Luc Florack, and Peter Johansen, editors, *Gaussian Scale-Space Theory*, chapter 11, pages 147–163. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[Der92] R. Deriche. Recursively Implementing the Gaussian and its Derivatives. In V. Srinivasan, Ong Sim Heng, and Ang Yew Hock, editors, *Proceedings of the 2nd Singapore International Conference on Image Processing*, pages 263–267. World Scientifi c, Singapore, 1992.

[DO00] John Dingliana and Carol O'Sullivan. Graceful degradation of collision handling in physically based animation. *Computer Graphics Forum*, 19(3), 2000.

[EL01] Stephan A. Ehmann and Ming C. Lin. Accurate and fast proximity queries between polyhedra using convex surface decomposition. In A. Chalmers and T.-M. Rhyne, editors, *EG 2001 Proceedings*, volume 20(3), pages 500–510. Blackwell Publishing, 2001.

[ES03] Kenny Erleben and Jon Sporring. Collision detection of deformable volumetric meshes. In Jeff Lander, editor, *Graphics Programming Methods*. Charles River Media, 2003.

[GBF03] E. Guendelman, R. Bridson, and R. Fedkiw. Nonconvex rigid bodies with stacking. *ACM Transaction on Graphics, Proceedings of ACM SIGGRAPH*, 2003.

[GDO00] Fabio Ganovelli, John Dingliana, and Carol O'Sullivan. Buckettree: Improving collision detection between deformable objects. In *Spring Conference in Computer Graphics (SCCG2000)*, pages pp. 156–163, Bratislava, April 2000.

[GJK88] E.G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Automation*, 4:193–203, 1988.

[GLM96] S. Gottschalk, M. C. Lin, and D. Manocha. Obb-tree: A hierarchical structure for rapid interference detection. Technical Report TR96-013, Department of Computer Science, University of N. Carolina, Chapel Hill, 8, 1996.

[Got00] Stefan Gottschalk. *Collision Queries using Oriented Bounding Boxes*. PhD thesis, Department of Computer Science, University of N. Carolina, Chapel Hill, 2000.

[GS87] Jeffrey Goldsmith and John Salmon. Automatic creation of object hierarchies for ray tracing. *IEEE Computer Graphics and Applications*, 7(5):14–20, May 1987. see Scherson & Caspary article for related work.

[He99] Taosong He. Fast collision detection using quospo trees. In *Proceedings of the 1999 symposium on Interactive 3D graphics*, pages 55–62. ACM Press, 1999.

[Hir02] Gentaro Hirota. *An Improved Finite Element Contact Model for Anatomical Simulations*. PhD thesis, University of N. Carolina, Chapel Hill, 2002.

[HKL+99] Kenneth E. Hoff, III, John Keyser, Ming Lin, Dinesh Manocha, and Tim Culver. Fast computation of generalized voronoi diagrams using graphics hardware. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 277–286. ACM Press/Addison-Wesley Publishing Co., 1999.

[Hub93] P. M. Hubbard. Interactive collision detection. In *Proceedings of the IEEE Symposium on Research Frontiers in Virtual Reality*, pages 24–32, 1993.

[Hub96] Philip M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Transactions on Graphics*, 15(3):179–210, 1996.

[Iij62] T. Iijima. Basic theory on normalization of a pattern (in case of typical one-dimensional pattern). *Bulletin of Electrotechnical Laboratory*, 26:368–388, 1962. (in Japanese).

[Kar] Karma. MathEngine Karma, http://www.mathengine.com/karma/.

[KHM+98] J. T. Klosowski, M. Held, J. S. B. Mitchell, H. Sowizral, and K. Zikan. Effi cient collision detection using bounding volume hierarchies of k-DOPs. *IEEE Transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998.

[Koe84] J. J. Koenderink. The Structure of Images. *Biological Cybernetics*, 50:363–370, 1984.

[KPLM98] S. Krishnan, A. Pattekar, M. Lin, and D. Manocha. Spherical shell: A higher order bounding volume for fast proximity queries. In *Proc. of Third International Workshop on Algorithmic Foundations of Robotics*, pages 122–136, 1998.

[Kui02] Arjan Kuijper. *The deep structure of Gaussian scale space images*. PhD thesis, Image Sciences Institute, Institute of Information and Computing Sciences, Faculty of Mathematics and Computer Science, Utrecht University, 2002.

[LAM01] Thomas Larsson and Tomas Akenine-Möller. Collision detection for continuously deforming bodies. In *Eurographics*, pages 325–333, 2001.

[LGLM99] Eric Larsen, Stefan Gottschalk, Ming C. Lin, and Dinesh Manocha. Fast proximity queries with swept sphere volumes. Technical Report TR99-018, Department of Computer Science, University of N. Carolina, Chapel Hill, 1999.

[LP90] L. M. Lifshitz and S. M. Pizer. A multiresolution hierarchical approach to image segmentation based on intensity extrema. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(6):529–541, 1990.

[Mel01] Stan Melax. Bsp collision detection as used in mdk2 and neverwinter nights. *Gamasutra*, March 2001. Online article.

[Mir98] Brian Mirtich. V-clip: Fast and robust polyhedral collision detection. *ACM Transactions on Graphics*, 17(3):177–208, July 1998.

[MW88] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *Computer Graphics*, volume 22, pages 289–298, 1988.

[OD99] C. O'Sullivan and J. Dingliana. Real-time collision detection and response using sphere-trees, 1999.

[Ode] Ode. Open Dynamics Engine, http://q12.org/ode/.

[Pal95] I.J. Palmer. Collision detection for animation: The use of the sphere-tree data structure. In *The Second Departmental Workshop on Computing Research*. University of Bradford, June 1995.

[PG95] I.J. Palmer and R.L. Grimsdale. Collision detection for animation using sphere-trees. *Computer Graphics Forum*, 14(2):105–116, 1995.

[PML97] M. K. Ponamgi, D. Manocha, and M. C. Lin. Incremental algorithms for collision detection between polygonal models:. *IEEE Transactions on Visualization and Computer Graphics*, 3(1):51–64, /1997.

[Set99] J. A. Sethian. Fast Marching Methods. *SIAM Review*, 41(2):199–235, 1999.

[SL00] K. Sundaraj and C. Laugier. Fast contact localisation of moving deformable polyhedras. In *IEEE Int. Conference on Control, Automation, Robotics and Vision*, Singapore (SG), December 2000.

[SSKJ03] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Software for Extracting Multi-Scale Singularity Trees. Technical report, Deliverable No.8, DSSCV, IST-2001-35443, 15. September 2003.

[SSKJ05] K. Somchaipeng, J. Sporring, S. Kreiborg, and P. Johansen. Extrema-Based Multi-Scale Singularity Trees: Soft-linked Scale-Space Hierarchies. In *Submitted to Scale-Space 2005*, 2005.

[TC96] C. Tzafestas and P. Coiffet. Real-time collision detection using spherical octrees : Vr application, 1996.

[vdB01] G. van den Bergen. Proximity queries and penetration depth computation on 3d game objects. *Game Developers Conference*, 2001.

[VM95] Pascal Volino and Nadia Magnenat Thalmann. Collision and self-collision detection: Effi cient and robust solutions for highly deformable surfaces. In Dimitri Terzopoulos and Daniel Thalmann, editors, *Computer Animation and Simulation '95*, pages 55–65. Springer-Verlag, 1995.

[VMT00] Pascal Volino and Nadia Magnenat-Thalmann. *Virtual Clothing, Theory and Practice*. Springer-Verlag Berlin Heidelbarg, 2000.

[Vor] CMLabs Vortex. http://www.cm-labs.com/products/vortex/.

[Wit83] A. P. Witkin. Scale–space fi ltering. In *Proc. 8th Int. Joint Conf. on Artificial Intelligence (IJCAI '83)*, volume 2, pages 1019–1022, Karlsruhe, Germany, August 1983.

[Zac98] G. Zachmann. Rapid collision detection by dynamically aligned dop-trees, 1998.