On constructing volume based approximation algorithms of spatial subsets

GÁBOR FÁBIÁN
Eötvös Loránd University
Department of Numerical Analysis
Pázmány Péter sétany 1/C.
BUDAPEST
robagnaibaf@gmail.com

LAJOS GERGÓ
Eötvös Loránd University
Department of Numerical Analysis
Pázmány Péter sétany 1/C.
BUDAPEST
gergo@inf.elte.hu

Abstract: In this paper we consider a general volume based approximation schema introduced in [6]. The schema allows us to generate convergent iterative processes to approximate polygonal (e.g. triangular) meshes. The approximation can be used for spatial decomposition or approximate convex decomposition. There are some parameters of the process, that can be choosed to fit it properly to the problem. Distance of meshes can be defined in several ways. Our goal is minimizing the error defined by the metric. The outline of the process is the following. The first approximation of the mesh is the bounding box. Thereafter every iterative step can be described like cutting by a plane. Since bounding box is convex polyhedron, after the first cut we obtain two convex polyhedra called atoms, moreover finitely many cuts result atoms, as well. Consequently, in the n-th step of the approximation we have a collection of $n+1$ atoms. The approximation is a mesh defined by the outer faces of some atoms, which have intersection with the input mesh. The most important questions are, how can we choose the atoms to be divided, how we define the dividing plane, and how we define the distance. In our research we test some choosing, dividing and distance functions to obtain some approximation processes. We can show, that most of the spatial decomposition processes are included in our general schema, therefore we hope, that we can find the best fitting approximation method for any approximation problem.

Key–Words: Volume based approximation, spatial decomposition, polyhedral approximation, convex decomposition

1 Introduction

This paper focuses on approximating algorithms of subsets of $\mathbb{R}^2$ or $\mathbb{R}^3$. As we will see, there are many possible applications of approximation algorithms of these sets. If the input set contains finitely many points, the approximation problem can be seen as a kind of space partitioning [12],[5]. If the input set is a polygonal mesh, then it can be considered as boundary of a spatial subset, then mesh repairing [1], mesh simplification, approximate convex decomposition [8],[14], can be obtained by the approximation process. Similarly, if input is a closed parametric surface, we can modify the process to obtain mesh generation or finite element methods [11].

The subjects mentioned before are very close to each other, our goal is to unify them into a single method, a general approximation algorithm. Different tuning of parameters fits various approximation problems. First, we give an overview of the process, thereafter we discuss its relation to other methods, finally we show a few examples about how should we tune the parameters to obtain different approximation methods.

2 Approximation schema

Let us consider $S \in \mathbb{R}^d$ an bounded subset of the plane or the space $(d = 2, 3)$. If S is bounded then it can be transformed into the d-dimensional unit ball, $\mathbb{B}^d$, i.e. we can assume, that $S \in \mathbb{B}^d$. The main idea is constructing atomic decomposition of the ball, which is obviously is a finite dimensional subspace of $\mathbb{B}^d$. By Riesz projection theorem we can draw up immediately the best approximation of any $S \in \mathbb{B}^d$, it is the Fourier-series with respect to the subspace spanned by the atoms [3],[9]. If the decomposition, $\mathcal{B}_0, \ldots, \mathcal{B}_n, \ldots$ is iteratively refined, we get a sequence of approximations $S_0, \ldots, S_n, \ldots$. In [6] we formulated some statements for the approximation sequence to be monotonic and convergent in a metric defined on ,,solid" sets of $\mathbb{B}^d$. Let us take a short survey of the mathematical details of the approximation schema:

Let $\mathcal{B}_n \subset 2^{\mathbb{B}^d}$ be a finite atomic σ-algebra on $\mathbb{B}^d$ [10], i.e. for $n \in \mathbb{N}$

$$\begin{aligned} &\mathcal{B}_n := \sigma(B_i^{(n)} \subset \mathbb{B}^d \ : \ i = 0, 1, \ldots, n) \\ &B_i^{(n)} \cap B_j^{(n)} = \emptyset \quad (i \neq j) \\ &\cup_{i=0}^{n} B_i^{(n)} = \mathbb{B}^d. \end{aligned} \tag{1}$$

Let μ be a measure on $\mathbb{B}^d$. It is easy to check [4],[13], that

$$\phi_i^{(n)} := \frac{1}{\sqrt{\mu(B_i^{(n)})}} \chi_{B_i^{(n)}} \quad (i = 0, 1, \ldots, n) \tag{2}$$

functions form orthonormed system under the common inner product of function spaces, $\langle ., . \rangle$. Here $\chi_{B_i^{(n)}}$ denotes the indicator function of the set $B_i^{(n)} \in \mathbb{B}^d$. Then for all f function the Fourier-series is defined by

$$\mathcal{F}^{\mathcal{B}_n} f := \sum_{i=0}^{n} \langle f, \phi_i^{(n)} \rangle \phi_i^{(n)}. \tag{3}$$

Consequently, the Fourier-series of an indicator function can be written as

$$\mathcal{F}^{\mathcal{B}_n} \chi_S := \sum_{i=0}^{n} \langle \chi_S, \phi_i^{(n)} \rangle \phi_i^{(n)} = \sum_{i=0}^{n} b_i^{(n)} \chi_{B_i^{(n)}}. \tag{4}$$

where

$$b_i^{(n)} := \frac{\mu(B_i^{(n)} \cap S)}{\mu(B_i^{(n)})}. \tag{5}$$

are the Fourier-coefficients. The main problem, that $\mathcal{F}^{\mathcal{B}_n} \chi_S$, is not an indicator function, therefore we can not assign an $S_n \subset \mathbb{B}^d$ to it. To solve the problem the following operator will be introduced:

$$\mathcal{X}_\alpha f(x) := \begin{cases} 0 & f(x) \leq \alpha \\ 1 & f(x) > \alpha \end{cases} \tag{6}$$

where $\alpha \in (0, 1)$. Now we can define the n-th approximation by

$$\begin{aligned} &S_n := \{ \mathcal{X}_\alpha \mathcal{F}^{\mathcal{B}} \chi_S = 1 \} = \\ &\{ x \in \mathbb{B}^d \mid \mathcal{X}_\alpha \mathcal{F}^{\mathcal{B}} \chi_S(x) = 1 \}. \end{aligned} \tag{7}$$

The distance of the sets $A, B \subset \mathbb{R}^d$ can be measured by

$$\rho(A, B) := \mu(A \triangle B) \tag{8}$$

where $\triangle$ denotes the symmetric difference. We showed, that ρ is a metric on bounded, connected, regular sets, which boundary is set of measure zero. Moreover, Lebesgue-density theorem implies that if $\mu(B_k^{(n)})$ tends to zero ($k \in (0, 1, \ldots, n)$ as n tends to infinity, then $\lim_{n \to \infty} \rho(S, S_n) = 0$, i.e. the sequence of approximations converges to S. Moreover the convergence is true, if only $\mu(B_k^{(n)})$ tends to zero, where $b_k^{(n)} \in (0, 1)$, i.e. atoms with Fourier-coefficients exactly 0 or 1 can be ignored. Let

$$\Delta_n := \{ i \in \{0, 1, \ldots, n\} \mid b_i^{(n)} \in (0, 1) \} \tag{9}$$

be the set of relevant indices, and let us define the diameter of a set B:

$$\operatorname{diam}(B) := \sup\{ \|x - y\|_2 \mid x, y \in B \}. \tag{10}$$

Then the convergence theorem can be formulated as follows.

Theorem 1

$$\lim_{n \to \infty} \max_{k \in \Delta_n} \operatorname{diam}(B_k^{(n)}) = 0$$

then

$$\lim_{n \to \infty} \rho(S_n, S) = 0.$$

To sum up, our input is an $S \in \mathbb{R}^d$ set, we need to define a measure μ on $2^{\mathbb{B}^d}$ (or at least outer measure), and choose an initial set $B_0^{(0)} \supset S$, let $n := 0$. Then

1. Choose an index k, where $0 < b_k^{(n)} < 1$, and divide the set $B_k^{(n)}$ into two non-empty disjoint sets, the resulting algebra is $\mathcal{B}_{n+1}$.
2. We can compute $\rho(S_n, S)$ using the measure (more accurately, the Fourier-coefficients). If $\rho(S_n, S) \geq \varepsilon$ for some given $\varepsilon > 0$ tolerance, go back to step 2, else the process terminates.

We need to answer two important questions. How can we choose the k index, and how we divide the $B_k^{(n)}$ atom. We introduce two functions $\mathcal{C}$ for choosing and $\mathcal{D}$ for dividing. $\mathcal{C}$ is a function from possible atomic decompositions to natural numbers, $\mathcal{D}$ maps any spatial subset to a pair of spatial subsets. Accurately, the following properties are required:

$$\forall n \in \mathbb{N} \quad 0 \leq \mathcal{C}(\mathcal{B}_n) \leq n \tag{11}$$

and

$$\begin{aligned} \forall B \in \mathbb{B}^d \quad &\mathcal{D}(B) = (B', B''), \\ &B' \cap B'' = \emptyset, \ B' \cup B'' = B. \end{aligned} \tag{12}$$

Using our notations the general approximation schema is the following.

VolumeBasedApproximation$(S, B_0^{(0)}, \varepsilon, \mathcal{C}, \mathcal{D})$

1. $n := 0$
2. $\mathcal{B}_n := \sigma(B_0^{(n)}, \dots, B_n^{(n)})$
3. $k := \mathcal{C}(\mathcal{B}_n)$
4. $S_n := \{\mathcal{X}_\alpha \mathcal{F}^{\mathcal{B}_n} \chi_S = 1\}$
5. **if** $\rho(S_n, S) \geq \varepsilon$ **then**
6. $\quad \mathcal{B}_{n+1} := \sigma(\dots, B_{k-1}^{(n)}, \mathcal{D}(B_k^{(n)}), B_{k+1}^{(n)}, \dots)$
7. $\quad n := n + 1$
8. $\quad$ **goto** 3.
9. $\quad$ **else stop**

(13)

3 Relation to space partitioning

In the followings let us consider, that the dividing operation is a splitting by a $d-1$ dimensional plane. This is not necessary, but in real world applications this is justifiable by the finite representation of spatial subsets. Therefore the spatial decomposition methods like quadtree, octree- k-d tree or BSP tree can be obtained naturally from the general schema. For example, we will present, how can we get a quadtree using our process. Let S be a finite subset of $\mathbb{B}^2$, and $B_0^{(0)}$ is the square of side 2 centered at the origin. Obviously $S \subset B_0^{(0)}$. We need to define the measure. If $B_k^{(n)}$ and S are disjoint sets, then $B_k^{(n)}$ is not relevant, therefore a good choice is the following:

$$\mu(B) := \begin{cases} 0 & B \cap S = \emptyset \\ \lambda(B) & otherwise \end{cases} \tag{14}$$

where λ denotes the common Lebesgue-measure, in this case the area of the B rectangle.

$$d_x(B) := \max\{|x' - x''| \mid (x', y), (x'', y) \in B\} \tag{15}$$

the diameter of B in the dircetion x, and similarly $d_y(B)$ the diameter of B in the direction y. Then let

$$J_n := \{j = 0, \dots, n \mid d_x(B_j^{(n)}) \neq d_y(B_j^{(n)}), \tag{16}$$

i.e. the set of indices of atoms which are not a square, and let

$$K_n := \{k = 0, \dots, n \mid \forall j \in \Delta_n \; \lambda(B_j^{(n)}) \leq \lambda(B_k^{(n)})\}, \tag{17}$$

i.e. the set of indices of atoms, that have greater or equal volume, than the relevant atoms. Then

$$\mathcal{C}(\mathcal{B}_n) := \begin{cases} \min J_n & J_n \neq \emptyset \\ \min K_n & otherwise \end{cases} \tag{18}$$

Thereafter, let us define $c_x(B) := \min_x\{x \mid (x, y) \in B\} + \frac{1}{2} d_x(B)$, the center of B in the direction x, and similarly $c_y(B)$ the center in the direction y. Then we can define the half-space

$$H_x(B) := \{(x, y) \in B \mid \langle (x, y) - c_x(B), (1, 0) \rangle \leq 0\}, \tag{19}$$

i.e. the points below the 1-dimensional plane defined by its normal $(1, 0)$ and a point c_x. Similarly, let

$$H_y(B) := \{(x, y) \in B \mid \langle (x, y) - c_y(B), (0, 1) \rangle \leq 0\}. \tag{20}$$

Then $\mathcal{D}$ can be defined as

$$\mathcal{D}(B) := (B', B'') \tag{21}$$

where

$$B' := \begin{cases} B \cap H_x(B) & d_x(B) \geq d_y(B) \\ B \cap H_y(B) & d_x(B) < d_y(B). \end{cases} \tag{22}$$

and $B'' = B \setminus B'$. It can be seen, that the approximation process with $\mu, \mathcal{C}, \mathcal{D}$ defined above generates the quadtree of the point set of S.

4 Decomposition

We assumed, that the initial atom of the decomposition is a convex polyhedron, which faces are convex polygons. Moreover, let us suppose that $\mathcal{D}$ operator is always can be interpreted as intersection with a half-space, so we have only $3 + 3$ free parameters, a normal vector and a point of the splitting plane. These limitations significantly simplify the problem. In this context, let us note some remarks (see e.g. [2]).

- The non-empty intersection of a convex polyhedron and a half-space is a convex polyhedron.
- If $B_0^{(0)}$ convex polyhedron and $\mathcal{D}$ is defined as splitting by a plane then $B_k^{(n)}$ is convex polyhedron for all $k, n \in \mathbb{N}$.
- The non-empty intersection of a convex polygon and a half-plane is a convex polygon.

- If the faces of $B_0^{(0)}$ are convex polygons and $\mathcal{D}$ is defined as splitting by a plane then the faces of $B_k^{(n)}$ are convex polyhedron for all $k, n \in \mathbb{N}$.

There are some additional consequences of the convexity of the faces of atoms.

Let us suppose, that p is a convex polygon defined by $v_1, \dots, v_n$ vertices, denote

$$p = [v_1 \dots v_n] := \operatorname{conv}(v_1, \dots, v_n) \tag{23}$$

the convex hull of the vertices. Let

$$P_{n,a} := \{x \in \mathbb{R}^3 \mid \langle x - a, n \rangle = 0\} \tag{24}$$

be a plane $(n, a \in \mathbb{R}^3)$, and let us suppose that $v_i \notin P_{n,a}$, i.e. the vertices of the polygon do not lay on the plane. If $p \cap P_{n,a} \neq \emptyset$ then exactly two segments are intersected, i.e. there exist exactly two indices i, j such that

$$[v_i v_{i+1}] \cap P_{n,a} \neq \emptyset, \quad [v_j v_{j+1}] \cap P_{n,a} \neq \emptyset. \tag{25}$$

i.e. there exist $x' \in [v_i v_{i+1}] \cap P_{n,a}$ and $x'' \in [v_j v_{j+1}] \cap P_{n,a}$. The intersecting points can be computed as follows

$$x' := v_i + \frac{\langle n, a - v_i \rangle}{\langle n, v_{i+1} - v_i \rangle}(v_{i+1} - v_i) \tag{26}$$

$$x'' := v_j + \frac{\langle n, a - v_j \rangle}{\langle n, v_{j+1} - v_i \rangle}(v_{j+1} - v_j) \tag{27}$$

Therefore, for a given atom the splitting operation can be done in few steps, namely: sort the faces below and above the splitting plane, collect the intersecting faces, divide the intersecting faces into two parts, find the caps of the intersecting surface. The main ideas of a polygonal mesh splitting algorithm can be found in [7]. In details the dividing operation is the following.

1. Find all intersecting faces. Compute all intersecting points. Every intersecting face has exactly two intersecting points x', x''.

2. Every $[v_1 \dots v_n]$ intersecting face will be divided into two non-empty faces

$$[x' x_i v_{j+1} \dots v_n v_1 \dots v_i] \tag{28}$$

and

$$[x' v_{i+1} \dots v_j x''] \tag{29}$$

using the notations of (25), (26) and (27), see Figure 1.

3. Let us suppose, that all the intersecting points $x'_1, x''_1, \dots, x'_L, x''_L$ are computed $(L \in \mathbb{N})$. Then let ν be a permutation of $\{1,\dots,L\}$, such that the intersecting vertices let be sorted by polar angle. It is easy to see, that $x'_{\nu_i} = x''_{\nu_{i+1}}$ or $x'_{\nu_{i+1}} = x''_{\nu_i}$ is true for all $i = \{1, \dots, L\}$, therefore the caps of intersecting surface can be obtained from the sorted vertices merging the equivalent elements.

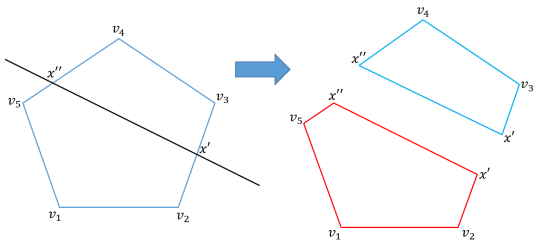

Figure 1: Splitting an intersecting polygon by a plane

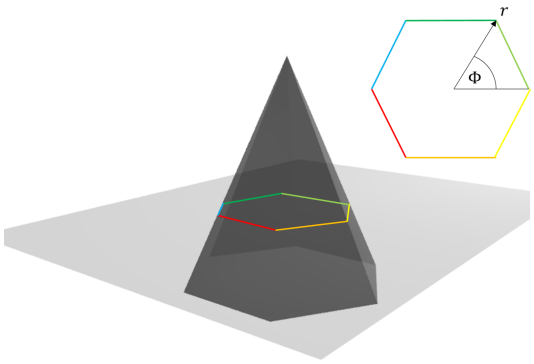

Figure 2: Caps of the intersecting surface can be computed by sorting vertices by polar angle.

The decomposition data structure contains three data arrays, namely vertices, faces and atoms. The array of vertices is an m-by-d matrix, where m is the number of vertices, the i-th row of the matrix contains the $(x, y, \dots)$ coordinates of the i-th vertex. Every element of the face array is an array of pointers pointing to vertices, accurately $f_i = (v^*_{i_1}, v^*_{i_2}, \dots, v^*_{i_{n_i}})$ means that f_i is a convex polygon defined by the vertices $v_{i_1}, \dots, v_{i_{n_i}}$, where $v^*_{i_j}$ is a pointer that points to v_{i_j}. Similarly, every element of the array of atoms is an array containing pointers to the faces. To sum up, decomposition is an array of atoms, which are convex polyhedra defined by convex polygons. The polygons are stored in a global face array, the vertices of the polygons in a global vertex array, in Figure 3-4. can be seen a simple example.

The decomposition is refined iteratively, in every step exactly one atom is chosen and split. There are many ways to export data from the data structure. It

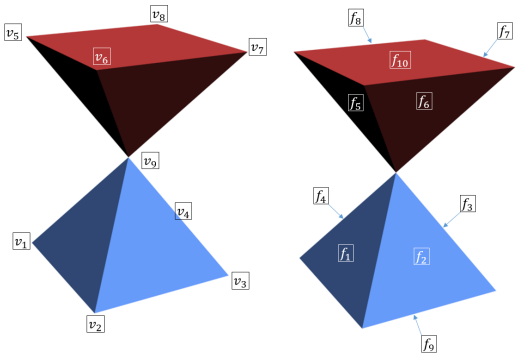

Figure 3: Decomposition with 2 atoms, 10 faces, 9 vertices. Both atoms have 5 faces, 4 triangles and 1 square. The v_9 vertex is shared between the atoms.

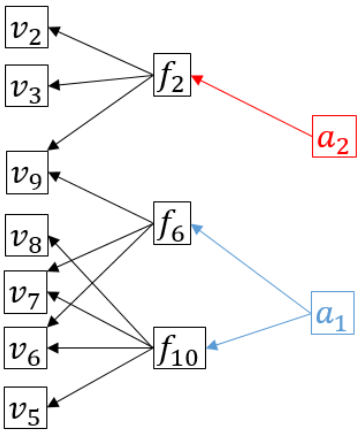

Figure 4: Some elements of the data structure of the decomposition shown in Figure 1.

is worth mentioning, that we can export every single atoms separately for e.g. fracture simulation, or we can export the boundary of the decomposition for e.g. mesh simplification. The boundary of the decomposition is defined by the faces of atoms, such that the face-adjacent atom is not a part of the approximation. So f_i is a boundary face if and only if it is a face of $B_k^{(n)}$ and $B_l^{(n)}$, moreover $b_k^{(n)} > \alpha$, but $b_l^{(n)} \leq \alpha$, see (7). If it is needed the polygonal mesh trivially can be converted to a simple triangular mesh, since all the polygons are convex.

5 Experimental Results

Without to claim the completeness, we show some results from our experiments. We used various choosing and dividing functions. Recall, that the output of the general approximation schema strongly dependent on these functions.

In every step of the iteration one atom is chosen, for example

- Random atom
- Maximum volume atom
- Maximum diameter atom
- Atom that containing the most vertices

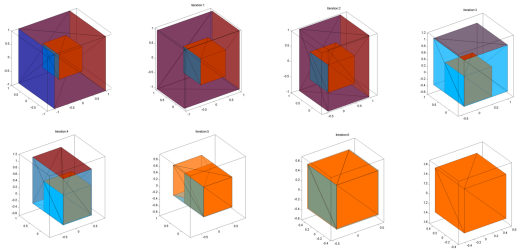

Figure 5: Approximation of a cube in 6 steps.

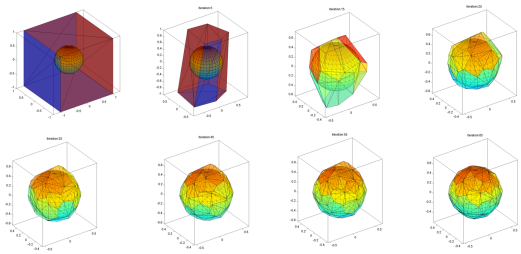

Figure 6: Approximation of a sphere in 65 steps.

Similarly, the parameters of the splitting plane can be defined in many ways. For example

- Random plane at the centroid of the atom
- Plane lying on a triangle contained in the atom
- Plane that least coplanar to any face of the atom
- Plane that best fitting to a part of the surface

Mesh approximation can be obtained in many ways, we tested our algorithm using various parametrization on some simple objects. Good example for the mesh simplification is a slowly varying surface represented with many vertices and faces. In this case our algorithm performs well even if many faces are missing from the input mesh.

The approximation schema with random plane splits can be used for fracture simulation of meshes, moreover it is possible to create BSP-like data structures.

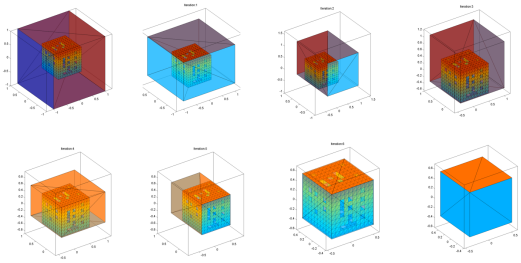

Figure 7: Approximation of a cube with topological errors in 6 steps. The initial mesh has $10 \times 10 \times 10\times$ segments (2 triangles per segment), several faces are removed from the surface.

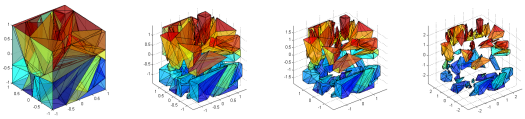

Figure 8: Fracture simulation using random planes.

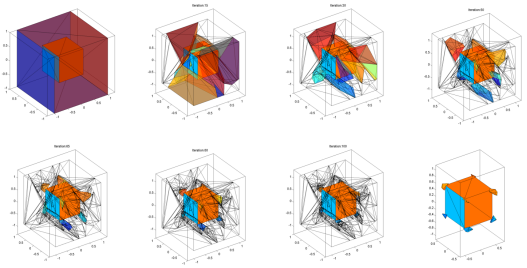

Figure 9: Space partitioning with random planes, the input is the 8 vertices of the cube.

6 Conclusions and Further Work

Our goal was to give a theoretical foundation for some similar approaches of space partitioning. The topics mentioned are often thought to be different, but all them can be described by a simple iterative approximation process. This approximation process is based on the theory of Fourier-series in Hilbert-spaces, which is well-known topic in mathematics, there are many useful theorems, that can be used during our work. But there are not just theoretical advantages of the method, we have a convergence theorem, necessary and sufficient condition for the monotonicity, and an exact error formula. If we can calculate exactly the volume of an intersection of an atom and the input, then we can calculate the exact distance, as well. Intersection of polyhedra can be calculated, but it is a difficult problem. Our idea is to calculate the volume of intersection without producing the intersection itself. If we had the exact error formula, and the Fourier-coefficients, we could optimize the performance of the approximation process, and maybe it would be more widely used.

References:

[1] M. Campen, M. Attene, L. Kobbelt *A Practical Guide to Polygon Mesh Repairing*, ACM Transactions on Graphics Vol. 21, Number 2, pp. 88-105 (2002)

[2] B. Chazelle, D. P. Dobkin *Intersection of Convex Objects in Two and Three Dimensions*, Journal of the ACM Vol. 34, Issue 1, pp. 1-27 (1987)

[3] E. W. Cheney *Introduction to Approximation Theory*, AMS Chelsea Pub. (1982)

[4] J. L. Doob *Measure Theory (Graduate Text in Mathematics)*, Springer (1994)

[5] C. Ericson *Real-Time Collision Detection*, Elsevier (2005)

[6] G. Fábián, L. Gergó. *Adaptive algorithm for polyhedral approximation of 3D solids*. Stud. Univ. Babes-Bolyai Math. Vol. 60, No. 2, 2015. pp 27591.

[7] G. Fabian, L. Gergo *Fast Algorithm to Split and Reconstruct Triangular Meshes*, Studia Universitatis Babes-Bolyai Series Informatica, Special Issue 1, pp. 90-102 (2014)

[8] M. Ghosh et al. *Fast approximate convex decomposition using relative concavity*, Computer-Aided Design Vol. 45, Issue 2, pp. 494-504 (2013)

[9] I. P. Natanson *Constructive Function Theory (Vol. I. Uniform Approximation)*, Frederick Ungar Publishing Co. (1964)

[10] J. Neveu *Discrete Parameter Martingales*, Elsevier (1975)

[11] S. J. Owen *A Survey of Unstructured Mesh Generation Technology*, International Meshing Roundtable, pp. 239-267 (1998)

[12] S. R. Tate, K. Xu *General-Purpose Spatial Decomposition Algorithms: Experimental Results*, Chapman&Hall/CRC (2004)

[13] M. E. Taylor *Measure Theory and Integration*, Graduate Studies in Mathematics, vol. 76 (2006)

[14] M. Muller N. Chentanez, T. Kim *Real Time Dynamic Fracture with Volumetric Approximate Convex Decompositions*, ACM Transactions on Graphics Vol. 32, Issue 4, pp. Article No. 115 (2013)