Technical Report: Virtual X-ray imaging for higher-order finite element results

Maximilian Bittens

Abstract

This work describes and demonstrates the operation of a virtual X-ray algorithm operating on finite-element post-processing results which allows for higher polynomial orders in geometry representation as well as density distribution. A nested hierarchy of oriented bounding boxes is used for preselecting candidate elements undergoing a ray-casting procedure. The exact intersection points of the ray with the finite element are not computed, instead the ray is discretized by a sequence of points. The element-local coordinates of each discretized point are determined using a local Newton iteration and the resulting densities are accumulated. This procedure results in highly accurate virtual X-ray images of finite element models.

1 Introduction

Finite element analyses in orthopaedic biomechanics provide deep insights into the physics of the human body. For such analyses to represent a meaningful complement to purely medical research, it can be useful to transfer the results into a medical imaging format, such as a radiographic or tomographic image. For example, in computational bone remodeling there are two major advantages to this approach: (1) the internal bone mineral density distribution is hard to infer from finite element post-processing results, since only the boundary of the specimen is immediately visible and (2) also people without technical background in finite elements, e.g. medicines, can evaluate the results in that way.

The simulation of X-ray images from a polygonal mesh is a well-researched topic and there are a number of simulation codes available, such as described in [1], [2] and [3], for example. [2] use Monte Carlo simulation to generate realistic images, while [1] and [3] use fast raycasting algorithms for that purpose. Whilst [3] account for density distribution with higher polynomial order, to the authors knowledge, there is no work available which operates directly onto finite element results and accounts for higher polynomial orders in both density distribution and geometry representation. For this purpose a novel approach will be presented in this approach.

2 Method description

In order to obtain an x-ray image of an arbitrary finite element result, the approved method of sending rays through the finite element mesh and subsequently integrating the quantity of interest along these rays will be adopted, here. Since an intersection test of each ray with every finite element would be computationally expensive, as a first step and for a given ray, the amount of finite element candidates for the intersection test have to be reduced. For this purpose, a tree-structure of nested bounding-boxes is employed, whereby the elements contained in a bounding box are bisected in the next hierarchical level. Since intersection tests with bounding boxes are very cheap, the tree of bounding boxes can be efficiently used to reduce the amount of candidate intersections.
elements. Since, even with this method, the determination of the exact points of intersection of the ray and the finite element will be computationally expensive if performed for a large amount of rays or a very large finite element mesh, numerical integration is introduced at this point. Thus, it is only necessary to test whether a discrete point is inside a finite element. As soon as the positional relationship of the point and the finite element is known in terms of element coordinates, it is simple to obtain the quantity of interest at the discrete point with the help of shape functions from a finite element postprocessing result.

3 Hierarchical-structured oriented bounding boxes

A bounding box for a collection of objects, such as an axis aligned bounding box (AABB) or an oriented bounding box (OBB) is a closed volume, that completely contains the collection of objects. In [4] a tree-structure of hierarchical OBB’s is introduced. Therein, principle component analysis [5] is used to create a set of nested tight-fitting bounding boxes. This method could also be used efficiently for contact detection in finite element method [4].
3.1 Principle component analysis

Let \( k = 1, \ldots, n \) and \( s_k \) be a linear triangle or so-called 2-simplex with vertices \( \hat{p}^k, \hat{q}^k \) and \( \hat{r}^k \) in \( \mathbb{R}^3 \). The centroid and the area of each simplex \( s_k \) in \( \mathbb{R}^3 \) may be calculated by

\[
\bar{c}^k = \frac{1}{3}(\hat{p}^k + \hat{q}^k + \hat{r}^k)
\]

and

\[
A^k = \sqrt{\gamma^k \left( \gamma^k - ||\hat{p}^k - \hat{q}^k|| \right) \left( \gamma^k - ||\hat{q}^k - \hat{r}^k|| \right) \left( \gamma^k - ||\hat{r}^k - \hat{p}^k|| \right)},
\]

subsequently, with

\[
\gamma^k = \frac{1}{2} \left( ||\hat{p}^k - \hat{q}^k|| + ||\hat{q}^k - \hat{r}^k|| + ||\hat{r}^k - \hat{p}^k|| \right).
\]

The (area-)weighted center of all simplices \( s_k \) can then be given by

\[
\mu = \frac{\sum k A^k \bar{c}^k}{\sum k A^k},
\]

which is independent of the distribution of the vertices in space. Using zero-centered centroids

\[
\bar{c}^k = \sqrt{A^k}(\bar{c}^k - \mu),
\]

the definition of the covariance matrix is given by

\[
C_{ij} = \frac{1}{n-1} \sum k c_i^k c_j^k,
\]

which is a symmetric positive (semi-)definite \( 3 \times 3 \)-matrix and thus owns real eigenvalues and, due to spectral theorem, an orthonormal basis \( B' \) in \( \mathbb{R}^3 \) consisting of the three eigenvectors out of which two point in the direction of maximum and minimum variance [5]. A change of basis \( T_B \) can be performed by

\[
T_B = (B')^{-1} B = (B')^{-1} I = (B')^{-1} = T_B',
\]

with identity matrix \( I \), since \( B \) is the standard basis for a Euclidean space. This procedure is easily extendable to higher dimensions or different primitives by specifying suitable definitions for \( c^k \) and \( A^k \) for the \( n \)-simplex with \( n > 2 \) or e.g. the \( n \)-cube.

3.2 Oriented bounding boxes

Let \( S = \{ s_k \}_{k=1, \ldots, n} \) be a set of 2-simplices or triangles and

\[
V = \bigcup_{s_k \in S} \{ \hat{p}^k, \hat{q}^k, \hat{r}^k \}
\]

the set of 0-simplices or vertices contained in \( S \). A convenient way to define a bounding box as an axis-aligned rectangular cuboid is given by the tuple

\[
B(V) = \left( P_{\text{min}}(V), P_{\text{max}}(V) \right), \quad \text{with} \quad P_{\text{min}}(V) = \left( \min_x(V), \min_y(V), \min_z(V) \right) \quad \text{and} \quad P_{\text{max}}(V) = \left( \max_x(V), \max_y(V), \max_z(V) \right).
\]

Obviously, \( P_{\text{min}} \) and \( P_{\text{max}} \) depend on the choice of the coordinate system. Choosing the standard
basis in $\mathbb{R}^3$ leads to the class of axis aligned bounding boxes (see figure 2). Allowing for different bases $B'$ an oriented bounding box

$$B_{B'}(V) = \left( P_{\min}(T_{B'}V), P_{\max}(T_{B'}V) \right),$$

(12)
can be defined.

Applying principle component analysis, shown in section 3.1, to the triangulation $T = (V, S)$ of the convex hull [6] of a body in $\mathbb{R}^3$ and using the obtained basis $B^T$ as an input for the orientated bounding box

$$B_{B^T}(V) = \left( P_{\min}(T_{B^T}V), P_{\max}(T_{B^T}V) \right),$$

(13)
the result will most likely fit the body tightly.

### 3.3 Hierarchical decomposition

In the latter, an algorithm was defined for the creation of tight-fitting OBBs around a body in $\mathbb{R}^3$. A top-down hierarchy is used for the generation of a series of nested OBBs with gradually smaller volume. Therefore, a parent OBB $B_{P}(\star)$ including the complete body is computed. The subdivision rule implemented then splits the bounding box with a plane orthogonal to the longest axis of the box in the center of mass. If the longest axis cannot be subdivided, the second longest axis is chosen. For the resulting two distinct regions two child OBBs $B^{C_1}_{C_1}(\star)$ and $B^{C_2}_{C_2}(\star)$ can be computed. This procedure may be repeated as long as a dividable convex hull of the fraction of the underlying geometry can be computed. The parent and child OBBs can be organized efficiently in a binary tree-structure. In figure 1 two steps of this procedure for AABBs and OBBs are shown using the example of an endoprosthesis. As can be seen, the OBBs converge faster than AABBs to the shape of the underlying geometry.

### 4 Raycasting geometric primitives in $\mathbb{R}^3$

In computer graphics the use of ray-surface or line-surface intersection tests, needed in a variety of related problems such as volume rendering, is termed raycasting [7]. A Ray can be defined by the tuple

$$R = (o, d), \quad o, d \in \mathbb{R}^3,$$

(14)
with the origin $o$ and the direction $d$ of the Ray.

In the following section the intersection of a ray with two different classes of geometric primitives is outlined: 1. the rectangular cuboid and 2. the tetrahedron.
4.1 Intersection of ray and rectangular cuboid

The class of rectangular cuboids is congruent with the class of oriented bounding boxes. Therefore we restrict ourselves to the determination of the intersection of a ray with an oriented bounding box. Via the map

\[ R^* = T_B \cdot R = (T_B \cdot o, T_B \cdot d) \]

(15)

the ray \( R \) can easily be transformed into the coordinate system of the OBB reducing the model problem to the collision of a ray \( R^* \) with an AABB. For this problem class, the slab method, first proposed in [8] is an easy and fast solution.

4.1.1 Slab method.

For the explanation of the slab-method the two dimensional example shown in figure 3 will be used. The procedure itself can be easily extended to three or more dimensions.

A ray \( R \) can also be expressed with following equation

\[ \vec{R}(t) = \vec{o} + t \cdot \vec{d}, \quad t \in \mathbb{R}^+, \]

(16)

which is a simple linear equation. The bounding box can be represented as the intersection of two slabs, the x-slab, defined by two parallel lines \( y = P_y^{\min} \) and \( y = P_y^{\max} \) both perpendicular to the y-axis, and the y-slab, defined by two parallel lines \( x = P_x^{\min} \) and \( x = P_x^{\max} \) both perpendicular to the x-axis. The intersection of the ray with the x-slab can be calculated by

\[ t_x^{\min} = \frac{(P_y^{\min} - o_y)}{d_y} \quad \text{and} \quad t_x^{\max} = \frac{(P_y^{\max} - o_y)}{d_y}. \]

(17)

Accordingly, the intersection of the ray with the y-slab is calculated by

\[ t_y^{\min} = \frac{(P_x^{\min} - o_x)}{d_x} \quad \text{and} \quad t_y^{\max} = \frac{(P_x^{\max} - o_x)}{d_x}. \]

(18)
The ray then intersects with the bounding box, if and only if the greatest minimum \( \max (t_{x_{\min}}^{\text{max}}, t_{y_{\min}}^{\text{max}}) \) is smaller than the smallest maximum \( \min (t_{x_{\max}}^{\text{max}}, t_{y_{\max}}^{\text{max}}) \), which holds true for \( R_1 \) and in contrast does not hold true for \( R_2 \), both depicted in figure 3. This algorithm allocates no memory and if \( \frac{1}{d} \) is pre-computed, it is division-free, in addition.

### 4.2 Intersection of ray and tetrahedron

A tetrahedron can be decomposed into its set of four 2-dimensional faces. Therefore we restrict ourselves to the intersection of a ray with a 2-simplex. For the determination of the intersection of a ray with a 2-simplex in \( \mathbb{R}^3 \) the Möller–Trumbore intersection algorithm [9] can be used as follows.

**Möller–Trumbore intersection algorithm.** A representation based on barycentric coordinates of a triangle \( s \) with vertices \( \hat{p}, \hat{q}, \) and \( \hat{r} \) can be given by the set of equations:

\[
s(u, v) = (1 - u - v) \cdot \hat{p} + u \cdot \hat{q} + v \cdot \hat{r}, \quad u, v \in \mathbb{R},
\]

\[
u \geq 0, \quad v \geq 0 \quad \text{and} \quad u + v \leq 1.
\]

(19)

The intersection of a ray \( R(t) \) with a triangle \( s(u, v) \) can be found by solving the equation

\[
R(t) = s(u, v) \Rightarrow o + t \cdot d = (1 - u - v) \cdot \hat{p} + u \cdot \hat{q} + v \cdot \hat{r}.
\]

(20)

Rearranging the terms

\[
-t \cdot d + u \cdot (\hat{q} - \hat{p}) + v \cdot (\hat{r} - \hat{p}) = o - \hat{p}.
\]

(21)

which can be seen as the translation of \( \hat{p} \) to the origin (see figure 4), yields the linear system of equations

\[
\begin{bmatrix}
-d, \hat{q} - \hat{p}, \hat{r} - \hat{p}
\end{bmatrix}
\begin{bmatrix}
t \\
u \\
v
\end{bmatrix}
= o - \hat{p},
\]

(22)

where \( M^{-1} \) can be seen as the transformation of \( R \) and \( s \), such that \( s \) is the unit-triangle shown in figure 4.

Denoting \( E_1 = \hat{q} - \hat{p}, \ E_2 = \hat{r} - \hat{p} \) and \( T = o - \hat{p} \) the solution to (22) can be obtained using Cramer’s rule

\[
\begin{bmatrix}
t \\
u \\
v
\end{bmatrix}
= \frac{1}{(d \times E_2) \cdot E_1}
\begin{bmatrix}
(T \times E_1) \cdot E_2 \\
(D \times E_2) \cdot T \\
(T \times E_1) \cdot d
\end{bmatrix}.
\]

(23)

The ray \( R(t) \) then intersects with the triangle \( s(u, v) \) at \( o + t \cdot d \), if and only if \( u > 0, v > 0, t > 0 \) and \( u + v \leq 1 \).
5 Positional relationship of a discrete point with a finite element

Finding the points of intersection for a ray and a finite element with higher polynomial order is not trivial and there are only a few sources available that provide a solution. Among them, [10] and [11] describe raycasting algorithms for high-quality visualization of finite elements. Since in this work we are not interested in visualizing finite elements the problem is reduced to the detection of the positional relationship of a discrete point \( \mathbf{x} \) and a finite element, as mentioned earlier. With this simplification the problem reduces to the search of the local coordinates \( \xi(\mathbf{x}) \) dependent on the point \( \mathbf{x} \), given in the global coordinate system.

5.1 Global-to-local iteration

For a given finite element with nodal coordinates \( \hat{\mathbf{x}}^i, i = 1, ..., n_{\text{nodes}}, \) associated shape functions \( N_i(\xi) \) and a point \( \mathbf{x} \), given in global coordinates, the element-local coordinates \( \xi \) need to be found, such that

\[
\mathbf{f}(\xi) = \mathbf{N}_i(\xi) \hat{\mathbf{x}}^i - \mathbf{x} = \mathbf{0}.
\]

This problem can be solved by the Newton-Raphson algorithm

\[
\xi^{(n+1)} = \xi^{(n)} - \mathbf{J}(\xi^{(n)})^{-1} \mathbf{f}(\xi^{(n)}),
\]

with \( \mathbf{J} \) being the Jacobian matrix with its components

\[
J_{ij} = \frac{\partial f_i}{\partial \xi_j}(\xi), \quad i = 1, 3, \quad j = 1, 3,
\]

an appropriate initial guess \( \xi_0 \) and a convergence criterion

\[
\frac{||\xi^{(n+1)} - \xi^{(n)}||}{||\xi^{(1)}||} < \varepsilon_{\text{tol}},
\]

with a user-defined tolerance \( \varepsilon_{\text{tol}} \).

5.2 In-hull test

Presuming \( \xi(\mathbf{x}) \) is known, the quantity of interest \( q \) can be interpolated in terms of shape functions from a finite element post-processing result as \( q(\mathbf{x}) = N_i(\xi(\mathbf{x})) \hat{q}^i. \) What remains is to check whether the local coordinates lie within the bounds of the reference element. Since all commonly used types of finite elements are usually convex, a simple test whether \( \xi(\mathbf{x}) \) lies within the convex hull of the reference finite element is sufficient. For the class of tetrahedral elements as used in this thesis an in-hull test, taking advantage of the barycentric coordinate system, can be performed by

\[
\sum_i \xi_i \leq 1 \quad \text{and} \quad \xi_i \geq 0, \quad i = 1, 3,
\]

for example.

6 Attenuation law

The initial intensity \( I_{in} \) of a beam of e.g. electromagnetic radiation decreases as it passes through a volume of matter. The Beer-Lambert law [12] relates this attenuation as

\[
I_{out} = I_{in} e^{-\int \mu(\mathbf{x}) d\mathbf{x}},
\]

where \( I_{out} \) is the resultant intensity and \( \mu \) is the linear attenuation coefficient. As an example, the linear attenuation coefficients for compact and cancellous bone are stated as 2.251 cm\(^{-1}\) and 0.716 cm\(^{-1}\), respectively, in [13].
7 X-ray generation

For the X-ray setup, a rather simple model is chosen here with an orthographic projection and monochromatic X-ray beams. In an orthographic projection the X-ray source is a plane with equally distributed parallel X-rays while monochromatic means each X-ray beam shares the same energy. For more information about possible X-ray simulation setups, the reader is referred to [3], for example.

A summary of the X-ray simulation process can then be stated as follows:

1. An axis aligned bounding box enclosing the complete finite element model is generated, whose surfaces can serve as emission plates.
2. A tree $G$ of hierarchical-structured oriented boundary boxes is generated to subdivide the finite element model.
3. For every X-ray $R_i$ sent through the model, the following steps are performed:
   (a) Find all leaves of $G$, which intersect with $R_i$.
   (b) Introduce numerical integration scheme $\bar{R}_{ij} = o + t_j d$ by sampling the ray equidistantly with a desired resolution.
   (c) For each discrete point $R_{ij}$, perform the global-to-local iteration only on the finite elements inside the leaf bounding boxes.
   (d) Perform numerical integration on the quantity of interest, e.g.

   $$I_{out} = I_{in} \exp \left[ - \sum_j w_j \mu(g(R_{ij})) \right].$$  \hspace{1cm} (30)

Albeit it is possible to introduce hierarchical levels in the tree $G$ until each leaf is filled only with one finite element, it is not always sensible to do so. It was found to be more efficient to create leaves with less than 10 elements and then order them for each ray $R_i$ by a linear guess performed by the Möller-Trumbore algorithm, introduced in 4.2.

8 Numerical examples

All methods described in this chapter were implemented within a self-developed framework, written in the Julia language [14]. In order to demonstrate the functionality of the method and the implementation, two numerical examples are derived here. The first one is a coarsely discretized ball with constant density and the second one is a finer discretized disk with a quadratic density distribution assigned to it.

8.1 Ball with constant density

As a first benchmark problem, a zero-centered ball $S: x^2 + y^2 + z^2 = r^2$, $x, y, z, r \in \mathbb{R}$ of radius $r = 1$ cm is discretized coarsely with 50 quadratic 10-node tetrahedral elements (see figure 5). A constant density of $\varrho = 1$ g/cm$^3$ is assigned to each element. An orthographic X-ray projection-plane is set up, as can be seen in figure 5, using the X-ray algorithm as an integration for the density of the discretized version of the ball. 23,668 rays per cm$^2$ are used to sample the discretized ball, which results in 94,864 rays in total, where the density is sampled along each ray with the same frequency, resulting in 29,218.112 sample points in total. In figure 6 the resulting X-ray image as well as the absolute error can be seen. The X-ray image looks as expected with a maximum density of 1.995 g/cm$^2$ at the middle of the picture, as the X-ray passes the full diameter here. Integration once again over the projected density results in a mass of 4.135 g, which has an error of approximately 1.3% error when compared to the analytically calculated
Figure 5: Ball discretized by 50 quadratic tetrahedral elements (left picture) and the model setup with ball, initial bounding box, X-ray projection plane and selected rays assembled in one scene (right picture).

mass \( m = \frac{4}{3}\pi r^3 \approx 4.189 \text{g} \). The absolute error, which can be seen as the sum of the discretization error of the ball and the discretization error of the rays, is measured at the location of each ray \( R_i \) by comparing the projected density with the integral \( \int_{r_i,\text{in}}^{r_i,\text{out}} \varrho \, dx \), where \( r_i,\text{in} \) and \( r_i,\text{out} \) are the intersection points of the ray with the analytical ball \( S \). It can be seen that the error is the smallest where the vertices of the 10-node tetrahedral elements are clipped to the surface of the ball, followed by the edges and finally the faces. As expected the largest error occurs at the edge of the projected circle. In addition concentric circles can be seen all over the picture, which can be interpreted as a sign of the depth discretization of the rays.

8.2 Cylinder with quadratic density distribution

As second test, a cylinder with radius \( r = 1 \text{cm} \) and height \( h = 0.1 \text{cm} \) is discretized with 2143 quadratic 10-node tetrahedral elements and a quadratic density distribution along the radius \( \varrho(r) = -4(r - 0.5)^2 + 2 \) is assigned to the model (both shown in figure 7). The X-ray setup is analogous to the latter example with the exception that there are less samples in the depth direction as the geometry and the density exhibit no variation in that direction and therefore the integration is exact even with just one sample along each ray.

In figure 8 the projected density and the absolute error can be seen. The projected density looks as expected, which once again demonstrates the function of the X-ray simulation. The absolute
Figure 7: Cylinder discretized by 2143 quadratic tetrahedral elements (left picture) and density distribution depending on the radius $r$ (right picture).

Figure 8: X-ray projection of the cylinder with density displayed in $\text{s/cm}^2$ (left picture) and plot of absolut error in $\text{s/cm}^2$ (right picture).

error is at least two magnitudes lower as in the first example. This can be explained by the simplified geometry, which, as already mentioned, exhibits exact integration per ray and by the finer mesh which renders the discretization of the cylinder almost exactly.
9 Bibliography

[1] FRED, N.; DUVAUCHELLE, P.; LÉTANG, J.M.; BABOT, D.: Fast and robust ray casting algorithms for virtual X-ray imaging. In: Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms 248 (2006), No. 1, pp. 175–180

[2] BARO, J.; SEMPAU, J.; FERNÁNDEZ-VAREA, J.M.; SALVAT, F.: PENEOPE: an algorithm for Monte Carlo simulation of the penetration and energy loss of electrons and positrons in matter. In: Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms 100 (1995), No. 1, pp. 31–46

[3] SUJAR, A.; MEULEMAN, A.; VILLARD, P.-F.; GARCIA, M.; VIDAL, F.: gVirtualXRay: virtual x-ray imaging library on GPU. In: Computer Graphics and Visual Computing The Eurographics Association, 2017, pp. 61–68

[4] GOTTSHALK, S.; LIN, M.C.; MANOCHA, D.: OBBTree: A hierarchical structure for rapid interference detection. In: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques ACM, 1996, pp. 171–180

[5] JOLLIFFE, I.: Principal component analysis. Springer, 2011

[6] BARBER, C.B.; DOBKIN, D.P.; DOBKIN, D.P.; HUHDANPAA, H.: The quickhull algorithm for convex hulls. In: ACM Transactions on Mathematical Software (TOMS) 22 (1996), No. 4, pp. 469–483

[7] PFISTER, H.; HARDEMBERGH, J.; KNITTEL, J.; LAUER, H.; SEILER, L.: The VolumePro real-time ray-casting system. In: Proceedings of the 26th annual conference on Computer graphics and interactive techniques, Association for Computing Machinery, 1999, pp. 251–260

[8] KAY, T.L.; KAJIYA, J.T.: Ray tracing complex scenes. In: ACM SIGGRAPH computer graphics Vol. 20(4) ACM, 1986, pp. 269–278

[9] MÖLLER, T.; TRUMBORE, B.: Fast, minimum storage ray/triangle intersection. In: ACM SIGGRAPH 2005 Courses ACM, 2005, pp. 7

[10] WILEY, D. F.; CHILDS, H.R.; HAMANN, B.; JOY, K.I.: Ray casting curved-quadratic elements. In: VisSym, 2004, pp. 201–210

[11] ÜFFINGER, M.; FREY, S.; ERTL, T.: Interactive High-Quality Visualization of Higher-Order Finite Elements. In: Computer Graphics Forum Vol. 29(2) Wiley Online Library, 2010, pp. 337–346

[12] SWINEHART, DF: The beer-lambert law. In: Journal of chemical education 39 (1962), No. 7, pp. 333

[13] SCHNEIDER, P.; BERGER, P.; MOLL, E.; REINERS, Chr; BÖRNER, W.: Getrennte Messung von Kompakta- und Spongiosadichte mit einem Transversal-Rotations-Scanner. In: RöFo-Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren Vol. 143(8) © Georg Thieme Verlag Stuttgart· New York, 1985, pp. 178–182

[14] BEZANSON, Jeff; EDELMAN, Alan; KARPINSKI, Stefan; SHAH, Viral B.: Julia: A fresh approach to numerical computing. In: SIAM review 59 (2017), No. 1, pp. 65–98