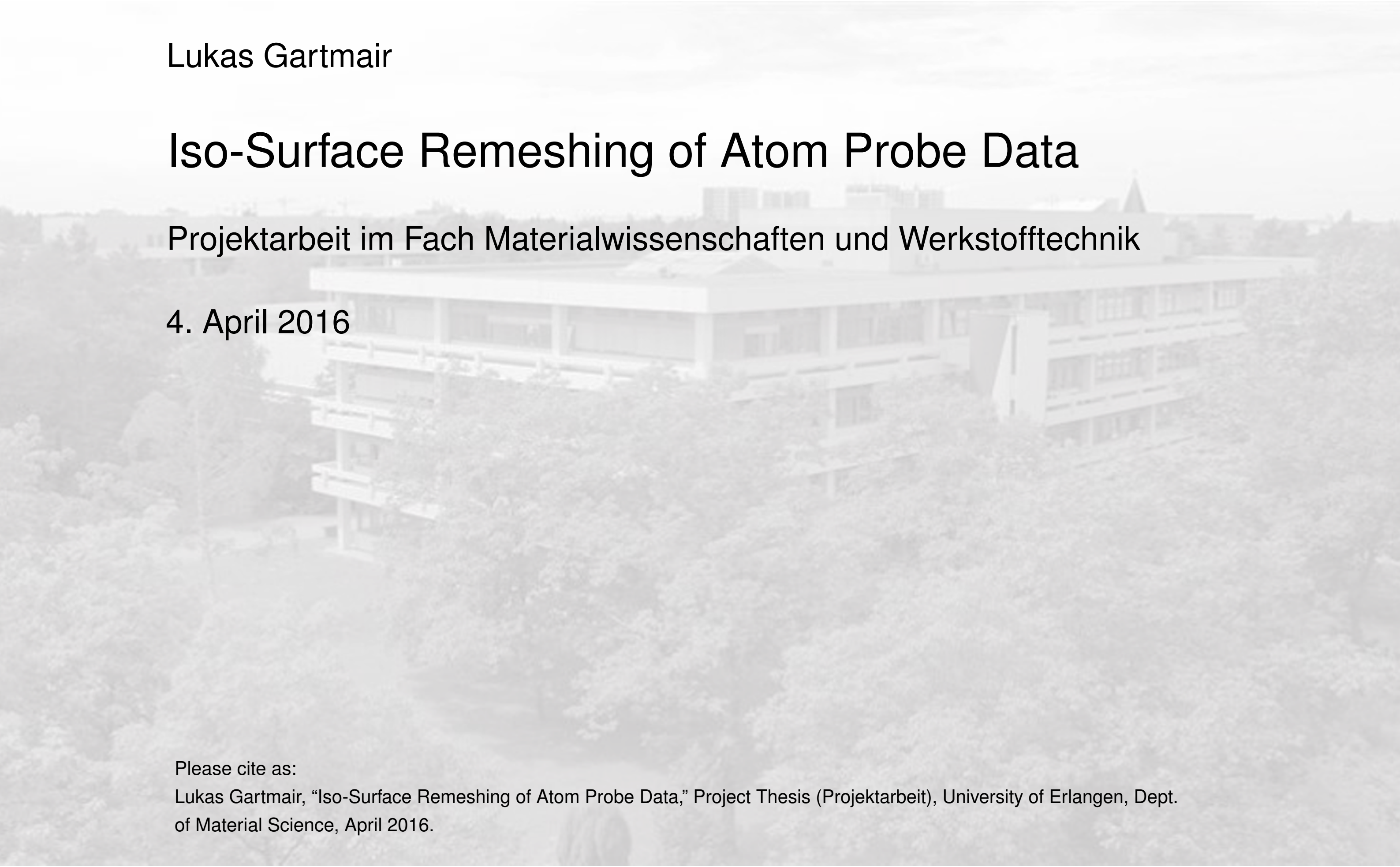
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Lehrstuhl WW1

Allgemeine Werkstoffeigenschaften



Lukas Gartmair

Iso-Surface Remeshing of Atom Probe Data

Projektarbeit im Fach Materialwissenschaften und Werkstofftechnik

4. April 2016

Please cite as:
Lukas Gartmair, “Iso-Surface Remeshing of Atom Probe Data,” Project Thesis (Projektarbeit), University of Erlangen, Dept. of Material Science, April 2016.

Friedrich-Alexander-Universität Erlangen-Nürnberg
Department Werkstoffwissenschaften
Allgemeine Werkstoffeigenschaften

Martensstr. 5 · 91058 Erlangen · Germany

http://www.gmp.ww.uni-erlangen.de/

Iso-Surface Remeshing of Atom Probe Data

Projektarbeit im Fach Materialwissenschaften und Werkstofftechnik

vorgelegt von

Lukas Gartmair

geb. am 05. January 1991
in München

angefertigt am

Lehrstuhl WW1
Allgemeine Werkstoffeigenschaften

Department Werkstoffwissenschaften
Friedrich-Alexander-Universität Erlangen-Nürnberg

Betreuer: **Prof. Dr. Peter Felfer**

Abgabe der Arbeit: **4. April 2016**

Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde.
Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.
I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

(Lukas Gartmair)
Erlangen, 4. April 2016

Contents

1 Introduction **1**

2 Motivation **2**

3 Concept **4**

4 Methods **5**

4.1 Voxelization 5

4.1.1 Dreamworks OpenVDB library 6

4.1.2 Adaptive treatment of sparse volumes 6

4.2 Contribution Transfer Function 7

4.2.1 Square Transfer Function 7

4.2.2 Sawtooth Transfer Function 8

4.3 Isosurface Extraction 9

4.3.1 Primal Methods 9

4.3.2 Dual Methods 10

4.3.3 Hybrids of Primal/Dual Methods 10

4.4 Voronoi Tesselation 12

4.4.1 Centroidal Voronoi Tesselation 13

4.5 Proximity Histogram and Surface excess of solute 15

5 Conclusion **17**

6 References **19**

Bibliography **21**

Chapter 1

Introduction

The characterization of nano-scaled materials becomes more and more important these days. This is due to the increasing interest of research in these materials and the downscaling of components in various fields of industries. The Atom Probe Tomography (APT) is an analysis technique that is able to resolve a material's structure on the atomic scale. It combines the methods Field Ion Microscopy (FIM) and Time of Flight Mass Spectrometer (TOFMS). A sharp tip, connected as positive electrode, becomes layerwisely stripped by the process of field evaporation. This effect is achieved by applying a high eletric field to the material causing *quantum tunneling* of the ionized atoms on the surface. A negatively biased counter electrode accelerates the detached ions on a two dimensional position detector. The species of each atom can be determined by measuring its time of flight. Thus it is possible to assign the detected positions to both a layer of the material and its element species. The third dimension can be reconstructed by subsequently compose the detected layers [1].

The spatial reconstruction of the measured data can be used to visualize the result in three dimensions in order to get further understanding of the experimental outcome.

Chapter 2

Motivation

The data measured in an Atom Probe Tomography experiment has to be visually and quantitatively interpreted. For this reason the collected data has to be visualized in such a manner that the final visualization represents the actual specimen information in the best way. Since the surface has to be calculated from the data gathered from the experiment the result is highly depending on the methods used for the calculation. Especially on the nanoscale, where ATP-measurements take place, even small deviations in the reconstruction may lead to a different understanding of the experiment.

Therefore the algorithms have to be chosen properly in order to minimize these deviations and the reconstruction artifacts.

The usage of common algorithms for both the extraction and the processing of the surface-mesh results in an isosurface with a sufficient precision. Nevertheless the potential to increase the precision by using more modern algorithms exists. The interpretation of ATP-data requires algorithms that provide an exact representation of detail. For example one of the most common algorithms to perform the isosurface extraction is the Marching Cubes (MC) algorithm [2]. But is has the major drawback that the actually existing sharp features are smoothed out and thus falsifies the actual shape.

The goal of this thesis is to overcome the existing drawbacks. Thus the visualization pipeline of the APT-analysis has to be improved in order to optimize the quality of the reconstruction. In particular the mesh of the generated isosurface. Quality here means to achieve the most accurate representation of the actual specimen information.

The proceeding is to create a visualization pipeline from scratch containing some selected algorithms and methods, which are performed consecutively. The goal is to write an APT data reconstruction tool using the open source 3D-Program *Blender* and *Python* as programming language.

Chapter 3

Concept

The visualization-concept of this thesis is to perform several tasks consecutively, which are listed below:

1. Voxelization of the raw data using a Contribution Transfer function
2. Isosurface Extraction
3. Centroidal Voronoi Tesselation - CVT
4. Calculation of proximity histograms

Each step is described further in the chapter Methods.

Chapter 4

Methods

4.1 Voxelization

The data resulting of APT-measurements consists of three-dimensional coordinates, belonging to the experimentally detected ions from the specimen [1]. Discretization of the arbitrary three-dimensional data simplifies its processing and calculation a lot. So the data is divided into uniform, three-dimensional cells, typically cubes [3], forming a regular grid. The cells with a certain unit volume are centered at the grid points [4]. These cells are called voxels (volume elements, cf. pixels in two dimensions) and are typically not stored by coordinates, but in a tree structure. Every voxel stores a value of a parameter describing the cells it represents [5, p. 127].

A schematic representation of voxels is shown in Figure 4.1.

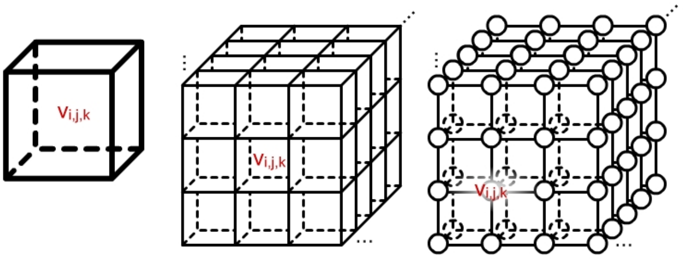


Figure 4.1 – Representation of voxels: a, single voxel b, voxel set c, voxel grid.

Source: http://johnrichie.com/V2/richie/isosurface/volume.html

4.1.1 Dreamworks OpenVDB library

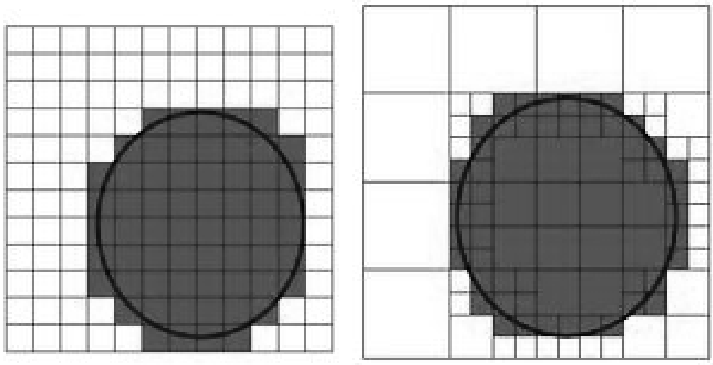
Regarding the great size of APT-data an efficient method for the discretization step is required. With the open source library OpenVDB written by K.Museth [6], Dreamworks provides a convenient solution for solving this problem. The key features of the library are:

- B+-tree based hierarchical data structure
- memory efficient
- no topology restrictions
- efficiency is on average(O(1))
- especially suitable for sparse volumes

For the storage of voxels in OpenVDB at compile time a fixed maximum tree height is chosen. The root node of the tree has a dynamic number of children at each node (branching factor). The internal nodes have fixed branching factors and the leaf nodes is limited to a fixed number of dimensions. In comparison to octrees and N-trees with fixed branching factors (2 and N) the branching factors of the OpenVDB structure can change within the tree levels. Their limitation is to take values that are powers of two. Only values that are stored in the leaf nodes correspond to individual voxels. These are the smallest addressable units [7].

4.1.2 Adaptive treatment of sparse volumes

Voxelization means creating a uniform, regular grid covering the whole volume. In many applications the volume is *sparse*, i.e. there are only few feature containing voxels while a huge amount of surrounding voxels is emptpy. Due to the great memory issues big data sets waste for the processing of featureless voxels the use of adaptive methods is a proper solution. Adaptive here means a more efficient representation of big data sets by using voxel sizes that convey the actual features [7]. The difference between a conventional and an adaptive rasterization (in three dimesions this is equal to voxelization) of a sparse volume is shown in Figure 4.2.



(a) conventional **(b)** adaptive

Figure 4.2 – Conventional and adaptive rasterization of a circle in sparse volume.

Source: http://media.bestofmicro.com/5/T/224417/original/raycasting-voxels

As the APT-data to be visualized is a highly sparse volume, the voxelization with the OpenVDB library is perfectly valid.

4.2 Contribution Transfer Function

The main problem with discretizing arbitrary volume data is that the measured ion positions do not match with the regular grid fitted in. This circumstance has to be taken into account by applying a transfer function. That means a weighted *'splatting'* of each atom to multiple surrounding grid points. The transfer functions as proposed by O.Hellman and D.Seidman [8] adress this problem.
The atom fractions contributing to each grid point are calculated with either a square or a sawtooth function. The problem of this proceeding is that every atom has to be either delocalized or displaced or both in order to be fitted on the grid.

The working principle of both the square and the sawtooth function are described in the following section.

4.2.1 Square Transfer Function

The square function assigns each atom to its closest grid point with full weight. The exact way how the contribution works is shown in Figure 4.3.

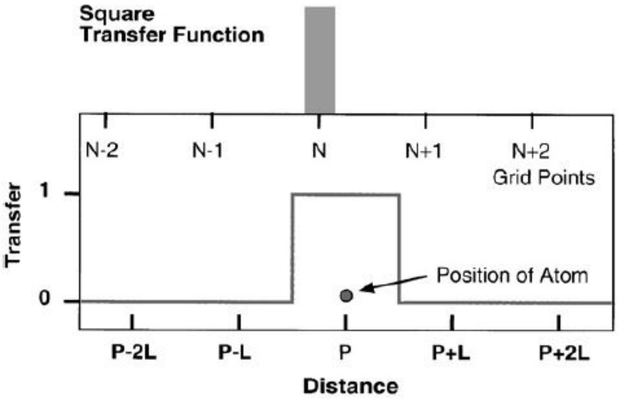


Figure 4.3 – Square Transfer Function

Source: [8]

An advantage of this method is of course the simplicity, but its disadvantages predominate. There is a spatial displacement from the atom to the grid point. Of course this causes a deviation from the original spatial configuration. In order to minimize this displacement error the grid would have to be infinitesimally small. But this causes an unreasonable calculation effort.

4.2.2 Sawtooth Transfer Function

A second option to spread each atoms density to the grid is the sawtooth function. The sawtooth function linearly distributes the contribution of an atom to its neighboring grid points. This contribution follows the lever principle, i.e. $the\,contribution \propto length\,of\ the\,opposite\,line\,segment$ (see figure 4.4).

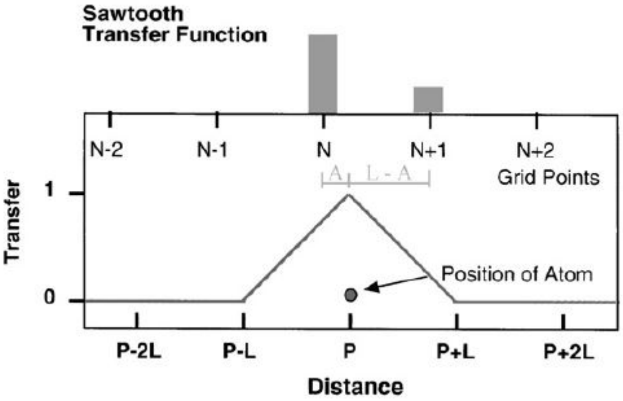


Figure 4.4 – Sawtooth Transfer Function

Source: [8]

The advantage is clearly that the original data becomes not spatially displaced. But every atom is splitted i.e. delocalized as there is no exact initial matching between the atom and any point of the grid.

But the delocalization of the atoms is not only a disadvantage. The increasing volume each atom contributes to after spreading improves the statistical representation.

4.3 Isosurface Extraction

The main aspect of this thesis is the extraction of isosurfaces. The process of converting the voxel grid into a mesh is named *Isosurface extraction* [9]. An isoconcentration surface in APT-data context describes a two dimensional space with the property that one atom species is present with a certain concentration (the so called isoconcentration-value). Also other isosurfaces like a surface with equal density or equal stoichiometry are common reconstructions of interest.

The way this step is performed has great influence on the resulting visualization. There are a lot of different methods that can be used in order to extract an isosurface from a discrete voxel grid, but with different resulting isosurfaces. Basically the methods are classified into Primal and Dual Methods. The difference between them is the way they locate the topology describing vertices [10].

4.3.1 Primal Methods

The approach of the primal methods is the following [10]:

1. Edges crossing the boundary become vertices in the isosurface mesh.
2. Faces crossing the boundary become edges in the isosurface mesh.
3. …
4. Cells crossing the boundary become (n-1)-cells in the isosurface mesh.

The most frequently used algorithm for this task is the Marching Cubes [2] [5], which is a primal method. The basic working principle is to walk iteratively through each cube of the grid and check the vertice signs in each cube. The vertex state is searched in a lookup-table and the related isosurface face is assigned to the current cube. The major drawback of MC is a ambiguity arising from different possibilities to connect the calculated vertices. Based on the MC algorithm lots of extensions were developed. An example grounding on MC is the Cubical Marching Squares (CMS) by [11].

4.3.2 Dual Methods

The second approach are Dual Methods. The dual methods follow this basic scheme [10]:

1. For every edge crossing the boundary, create an (n-1) cell. (Face in 3D)
2. For every face crossing the boundary, create an (n-2) cell. (Edge in 3D)
3. ...
4. For every d-dimensional cell, create an (n-d) cell.
5. ...
6. For every n-cell, create a vertex.

A famous representative of dual methods is the SurfaceNets algorithm by [12]. Here the first step is to check the grid for edges that contain a sign change. In three dimensions every edge belongs to four cubes. So the connection of the four cubes' representing vertices results in a quad.

The relationship between MC and SurfaceNets is that they share the primal/dual property. I.e. the vertices of the SurfaceNets mesh are associated with the faces of the MC's mesh. The same is valid the other way around. [13]

4.3.3 Hybrids of Primal/Dual Methods

There is another alternative that takes the advantages of both a cube-based method and a dual method. An example for this approach is the Extended Marching Cubes (EMC) [14]. The special thing about this method is to take the 'normals associated with the intersection points on the edges of a cube' [13] into consideration. This proceeding allows to detect sharp features inside the current cube. Features are described by a defined cone. Featureless cubes are treated with standard MC, the vertices of cubes containing features are generated by minimizing the quadratic function:

$$E[x] = \sum_i (n_j \cdot (x - p_i))^2 \tag{4.1}$$

Based on EMC and SurfaceNets algorithm Ju et al. [13] proposed a new method called '*Dual Contouring*'. It runs basically two steps for vertex generation and connecting them to surface simplices.

1. For each cube that exhibits a sign change, generate a vertex positioned at the minimizer of Equation (4.1) (cf. EMC).
2. For each edge that exhibits a sign change, generate a quad connecting the minimizing vertices of the four cubes containing the edge (cf. SurfaceNets).

This proceeding has the great advantage that there is no need to specifically test for features. This algorithm requires the presence of Hermite Data, i.e. for every edge in the voxel grid containing a sign change the intersection point of the contour is known. In addition to this the gradient (the first partial derivations) of the contour has to be calculated. As the gradient is perpendicular to the contour it can be interpreted as the normals of the isosurface.

This is schematically shown in Figure 4.5.

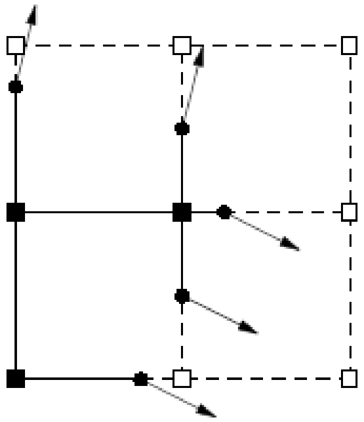


Figure 4.5 – Hermite data (Intersection points and its gradients) on a signed grid.

Source: [13]

The resulting positioning of vertices with Dual Contouring (DC) based on the Hermite Data from above is shown in figure Figure 4.6

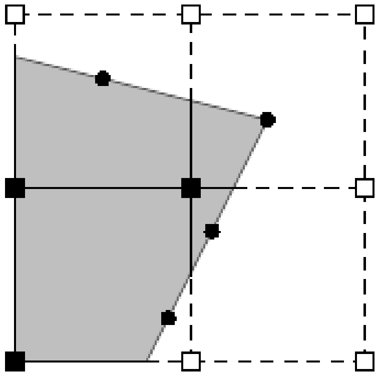


Figure 4.6 – Vertex generation with the dual contouring method.

Source: [13]

There are several other approaches on this problem. Without detailed explanation these are listed below.

- Dual Marching Cubes (DMC) [15]
- Adaptive Skeleton Climbing (ASC) [16]

4.4 Voronoi Tesselation

The Voronoi Tesselation describes the segementation of a plane into regions (Voronoi cells) with the property that every cell describes the shortest euclidean distance to its generator (a point in the plane) [17]. This feature is mathematically expressed in Equation (4.2).

$$\widehat{V_i} = \left\{ x \in \Omega \mid |x - g_i| < |x - g_j| \, for \; j = 1, \ldots, k, \; j \neq i \right\} \tag{4.2}$$

In Figure 4.7 an exemplary Voronoi Tesselation is shown with 20 generators and the corresponding centroids of the Voronoi cells.

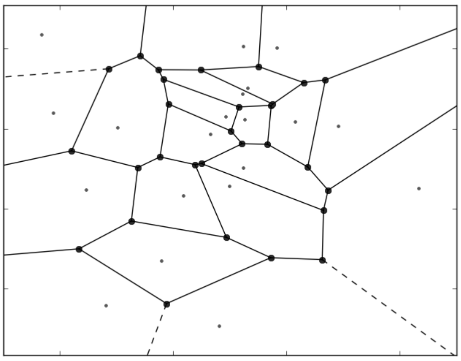


Figure 4.7 – Exemplary Voronoi tesselation with 20 generators and the gaussian normal distribution as density function.

The dual graph of the Voronoi Tesselation is the Delaunay Triangulation since the Voronoi generators are equal to the triangle vertices of the Delaunay tesselation and vice versa (see Figure 4.8). The core principle of graph duality has already been discussed in Section 4.3.

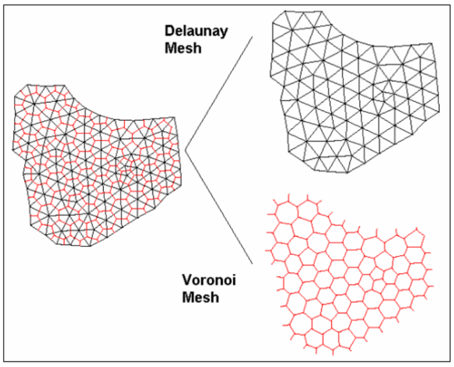


Figure 4.8 – Dual relationship between the Voronoi and the Delaunay tesselation.

Source: [18]

4.4.1 Centroidal Voronoi Tesselation

A special case of the Voronoi Tesselation is the Centroidal Voronoi Tesselation (CVT). In this case all generators g coincide with the position of the mass centroid g^* of each cell corresponding to a given density function. The calculation of the mass centroid is expressed in Equation (4.3) [17].

$$g^* = \frac{\int_V y \cdot \rho(y)\, \mathrm{d}y}{\int_V \cdot \rho(y)\, \mathrm{d}y} \tag{4.3}$$

The CVT has the advantage over the conventional Voronoi that its is more equally distributed and structured (see Figure 4.9 and Figure 4.10).

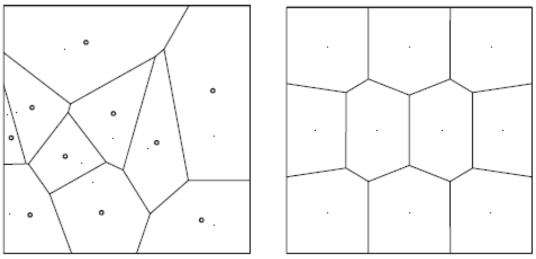


Figure 4.9 – Comparison between a 10 generator Voronoi diagram and its centroidal analogue with the density function $\rho(y) = const.$.

Source: [17]

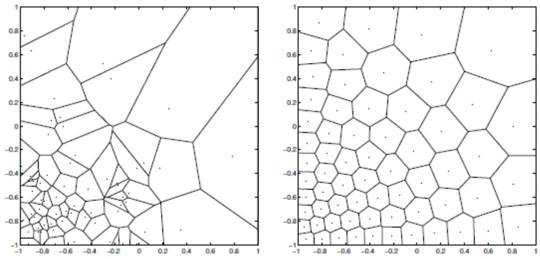


Figure 4.10 – Comparison between a 64 generator Voronoi diagram and its centroidal analogue with the density function e^{-2x-2y}.

Source: [17]

This is an important feature assuming the Voronoi Tesselation to be a mesh that represents a certain geometry. In such cases the equal distribution property can be very useful as it optimizes the mesh quality for further mesh processing techniques such as visualization [19]. Furthermore the mesh optimization becomes much more robust against initial mesh configuration and the distribution of the vertices. Comparable optimization techniques like the Laplacian Smoothing are more sensitive to these conditions [20].

A common algorithm used to obtain a CVT from a Voronoi Tesselation is the Lloyd's relaxation by [21]. The basic working principle is to iteratively repeat the following two steps until a sufficient approximation to the CVT is reached:

1. Build the Voronoi Tesselation corresponding to the mesh vertices

2. Computation of the centroids according to the given density function (cf. Equation (4.3)) and replace the site of each cell with the calculated centroid of the cell

4.5 Proximity Histogram and Surface excess of solute

The surface excess of solute [22] is an important thermodynamic property of interfaces like grain boundaries and precipitation/bulk boundaries. It takes the adsorption of atoms on the boundary into account, which leads to a higher concentration of a component on the boundary than in the bulk. By comparing the amounts of substance of the real behavior to an ideal reference system the amount of substance on the interface can be determined. Out of this by dividing n by the area of the interface the areal surface excess concentration can be calculated [23].

The problem with this calculation is that in reality interfaces are rarely smooth and thus the determination of the area becomes complicated.

Helman et al. [8] proposed a method to calculate the surface excess without the need to mathematically determine the area of the interface. The Proximity histogram (a so called proxigram) describes the concentration of an atom species in relation to the distance of a isoconcentration surface. Their method bases on the analysis of proximity shells around the isosurface. The analysis is based on counting the atoms inside this specified range. With the number of atoms and the known material density the local concentration inside the shell can be calculated (see Figure 4.11).

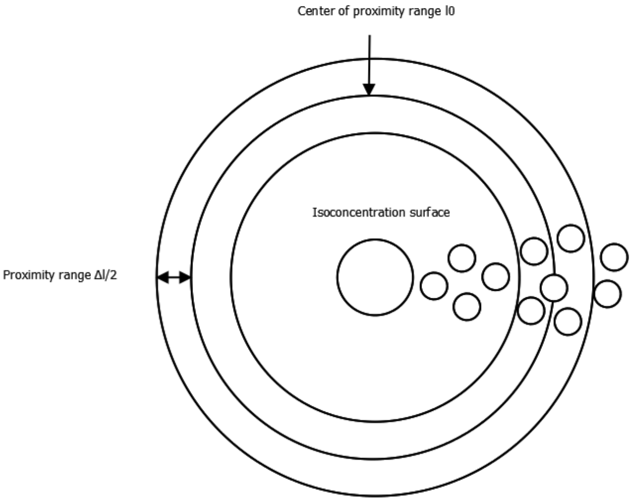


Figure 4.11 – Schematic proximity shell with center l_0 and range $\delta l/2$. The number of atoms N_l within the range here is 5. The total atom number is 11.

Repeating this for subsequent proximity ranges results in a function of concentration over the distance to the isosurface (proximity). An example is shown in Figure 4.12.

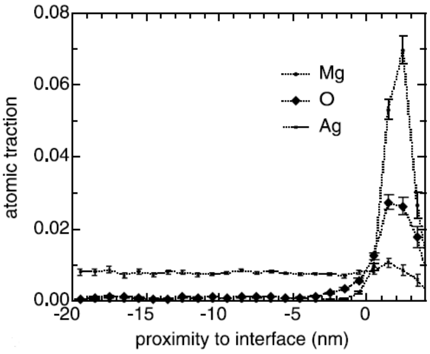


Figure 4.12 – Exemplary Proxigram.

Source: [8]

A positive distance value describes a position inside the isosurface. Proxigrams are able to reveal the actual position of the isosurface to the interface by comparing the zero point (position of the isosurface) with the concentration peaks (position of an atom species).

Chapter 5

Conclusion

In the previous sections some concepts were described, which all are tools for developing a consistent analysis tool.
That is the process of a memory efficient voxelization that simplifies working and processing with the data. Secondly there is the main task, the extraction of iso-surfaces. This step requires the most attention as it is the core problem to solve and has the greatest potential of improvement. At last there are some analytical methods in order to gain quantitative information from the experiment. These are the calculation of proximity histograms and out of that the determination of the surface excess of solute.

List of Acronyms

FIM	Field Ion Microscopy
APT	Atom Probe Tomography
TOFMS	Time of Flight Mass Spectrometer
MC	Marching Cubes
APT	Atom Probe Tomography
CMS	Cubical Marching Squares
EMC	Extended Marching Cubes
DC	Dual Contouring
DMC	Dual Marching Cubes
ASC	Adaptive Skeleton Climbing
CVT	Centroidal Voronoi Tesselation

Chapter 6

References

List of Figures

4.1 Representation of voxels: a, single voxel b, voxel set c, voxel grid. . 5
4.2 Conventional and adaptive rasterization of a circle in sparse volume. 7
4.3 Square Transfer Function 8
4.4 Sawtooth Transfer Function 8
4.5 Hermite data (Intersection points and its gradients) on a signed grid. 11
4.6 Vertex generation with the dual contouring method. 12
4.7 Exemplary Voronoi tesselation with 20 generators and the gaussian normal distribution as density function. 13
4.8 Dual relationship between the Voronoi and the Delaunay tesselation. 13
4.9 Comparison between a 10 generator Voronoi diagram and its centroidal analogue with the density function $\rho(y) = const.$. 14
4.10 Comparison between a 64 generator Voronoi diagram and its centroidal analogue with the density function e^{-2x-2y}. 14
4.11 Schematic proximity shell with center l_0 and range $\delta l/2$. The number of atoms N_l within the range here is 5. The total atom number is 11. 16
4.12 Exemplary Proxigram. 16

Bibliography

[1] O. Hellman, J. Vandenbroucke, J. B. du Rivage, and D. N. Seidman, “Application software for data analysis for three-dimensional atom probe microscopy,” *Materials Science and Engineering: A*, vol. 327, no. 1, pp. 29 – 33, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921509301018871

[2] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3d surface construction algorithm,” *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, Aug. 1987. [Online]. Available: http://doi.acm.org/10.1145/37402.37422

[3] S. Gebhardt, E. Payzer, and L. Salemann, “Polygons, point-clouds, and voxels, a comparison of high-fidelity terrain representations,” 06.

[4] S. W. Wang and A. E. Kaufman, “Volume-sampled 3d modeling,” *IEEE Comput. Graph. Appl.*, vol. 14, no. 5, pp. 26–32, Sep. 1994. [Online]. Available: http://dx.doi.org/10.1109/38.310721

[5] C. Johnson and C. Hansen, *Visualization Handbook*. Orlando, FL, USA: Academic Press, Inc., 2004.

[6] K. Museth, “Vdb: High-resolution sparse volumes with dynamic topology,” *ACM Trans. Graph.*, vol. 32, no. 3, pp. 27:1–27:22, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2487228.2487235

[7] “Openvdb - faq,” http://www.openvdb.org/documentation/doxygen/faq.html, accessed: 2015-05-01.

[8] O. Hellman and D. Seidman, “Measurement of the gibbsian interfacial excess of solute at an interface of arbitrary geometry using three-dimensional atom probe microscopy,” *Materials Science and Engineering: A*, vol. 327, no. 1, pp. 24–28, 2002.

[9] E. S. Lengyel, “Voxel-based terrain for real-time virtual simulations,” Ph.D. dissertation, Davis, CA, USA, 2010, aAI3404919.

[10] M. Lysenko, "Smooth voxel terrain," http://0fps.net/2012/07/12/smooth-voxel-terrain-part-2/, July 2012, accessed: 2015-05-01.

[11] C. chang Ho, F. che Wu, B. yu Chen, and M. Ouhyoung, "Cubical marching squares: Adaptive feature preserving surface extraction from volume data," *Computer Graphics Forum*, vol. 24, p. 2005, 2005.

[12] S. Gibson, "Constrained elastic surface nets: Generating smooth surfaces from binary segmented data," in *Medical Image Computing and Computer-Assisted Interventation 98*, ser. Lecture Notes in Computer Science, W. Wells, A. Colchester, and S. Delp, Eds. Springer Berlin Heidelberg, 1998, vol. 1496, pp. 888–898. [Online]. Available: http://dx.doi.org/10.1007/BFb0056277

[13] T. Ju, F. Losasso, S. Schaefer, and J. Warren, "Dual contouring of hermite data," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 339–346, Jul. 2002. [Online]. Available: http://doi.acm.org/10.1145/566654.566586

[14] S. Raman and R. Wenger, "Quality isosurface mesh generation using an extended marching cubes lookup table," *Computer Graphics Forum*, vol. 27, no. 3, pp. 791–798, 2008. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2008.01209.x

[15] S. Schaefer and J. D. Warren, "Dual marching cubes: Primal contouring of dual grids." in *Pacific Conference on Computer Graphics and Applications*. IEEE Computer Society, 2004, pp. 70–76. [Online]. Available: http://dblp.uni-trier.de/db/conf/pg/pg2004.html#SchaeferW04

[16] T. Poston, T.-T. Wong, and P.-A. Heng, "Multiresolution isosurface extraction with adaptive skeleton climbing," *Computer Graphics Forum*, 1998.

[17] Q. Du, V. Faber, and M. Gunzburger, "Centroidal voronoi tessellations: Applications and algorithms," *SIAM Rev.*, vol. 41, no. 4, pp. 637–676, Dec. 1999. [Online]. Available: http://dx.doi.org/10.1137/S0036144599352836

[18] D. Swenson, S. Gosavi, and B. Hardeman, "Integration of poroelasticity into tough2," in *Twenty-Ninth Workshop on Geothermal Reservoir Engineering*. Stanford University, January 2004.

[19] P. Alliez, É. Colin De Verdière, O. Devillers, and M. Isenburg, "Centroidal voronoi diagrams for isotropic surface remeshing," *Graphical Models*, vol. 67, no. 3, pp. 204–231, 2005. [Online]. Available: https://hal.inria.fr/hal-00787166

[20] D. Wang and Q. Du, "Mesh optimization based on the centroidal voronoi tesselation," *INTERNATIONAL JOURNAL OF NUMERICAL ANALYSIS AND MODELING*, vol. 2, pp. 100–113, 2005.

[21] S. P. Lloyd, “Least squares quantization in pcm.” *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–136, 1982. [Online]. Available: http://dblp.uni-trier.de/db/journals/tit/tit28.html#Lloyd82

[22] J. Gibbs, *The Scientific Papers of J. Willard Gibbs*, vol. 1, pp. 219–331, 1961.

[23] A. C. Mitropoulus, “What is a surface excess,” *Journal of Engineering Science and Technology Review 1*, vol. 1-3, 2008.