Fault-Tolerant Defect Prediction in High-Precision Foundry

Mikel Salazar, David Gallego, Yoseba K. Penya, Igor Santos, Pablo G. Bringas
S^3lab, Deusto Technology Foundation, Bilbao, Spain
Email: {msalazar, dgallego, ypenya, isantos, pgb}@tecnologico.deusto.es

***Abstract*—High-precision foundry production is subjected to rigorous quality controls in order to ensure a proper result. Such exams, however, are extremely expensive and only achieve good results in *a posteriori* fashion. In previous works, we presented a defect prediction system that achieved a 99% success rate. Still, this approach did not take into account sufficiently the geometry of the casting part models, resulting in higher raw material requirements to guarantee an appropriate outcome. In this paper, we present here a fault-tolerant software solution for casting defect prediction that is able to detect possible defects directly in the design phase by analysing the volume of three-dimensional models. In order to accomplish this goal, we propose advanced algorithms to recreate the topology of each foundry part, analyze its volume and simulate the casting procedure, all of them specifically designed for a robust implementation over the latest graphic hardware that ensures an interactive design process.**

I. Introduction

High-precision foundry has become the mainstay of key industries in modern times producing core components present in our everyday lives, from razor blades to the latest piece of aeronautical equipment. Since the tiniest defect in final foundry pieces may be fatal, there are very strict quality standards to assure a minimum quantity. Still, these controls are performed when the production is already finished, with the subsequent cost increment.

The goal of foundries nowadays is to achieve zero defects in all castings but this objective is extremely difficult to accomplish due to the huge amount of variables involved in the foundry process (e.g. composition, size of the casting, cooling speed or thermal treatment). All these factors influence the quality of the final product and must be taken into account in the design phase.

In previous works, we conducted a research focused on two of the most difficult to avoid and extended defects, namely micro-shrinkages and incongruities in mechanical property values [1], [2], [3]. Although we managed to top the defect prediction rate up to the 99%, the results were negatively affected by the geometric complexity of the foundry piece, since we used basic numeric values instead of the complete three-dimensional representation.

As a consequence of the high requisites associated with modern foundry production, casting part definition requires complex modelling paradigms to describe the exact geometry of the piece, Constructive Solid Geometry(CSG) being the most widely used. This modelling technique allows designers to define tridimensional solid objects by hierarchically combining simple geometric primitives using boolean operators and basic transformations [4].

Due to the nature of the foundry processes, casting part design relies heavily on volume-based geometric reasoning. Volumetric data provides a better understanding of the internal structure of the model and must be obtained through a process know as voxelization. This kind of algorithm extract a discrete representation of the volumetric information from the part model. The resulting data is stored into a tree-dimensional matrix in which each element is called a voxel (a linguistic blend of the words *volumetric* and *pixel*).

In recent years, the evolution of graphics hardware has given rise to faster and more accurate voxelization algorithms [9], [8], [10], [11]. Thanks also to the emergence of programmable shaders (sets of code instructions executed directly on the graphic processing units), much of the computational burden has been shifted to specialized hardware, so it is now possible to conduct voxelization and simulation operations in real-time [12], [13].

Unfortunately, there are currently few casting process simulation solutions that take advantage of those volumetric analysis techniques and, furthermore, those require that the three-dimensional models be extremely well defined, forcing the designer to use complicated surface healing procedures and, more often than not, to manually reconstruct unmatched surfaces, which is a difficult, tedious and time-consuming process.

Against this background, we present here a fast, flexible, and fault-resistant software solution that manages to remove all the casting defects already in the design phase. With this solution, we advance the state of the art in two ways. First, we provide a new method to obtain the volumetric space within a foundry piece model, even if there are errors in its definition. Second, we improve the simulation and visualization of foundry processes through computational geometry and computer graphics techniques such as straight line skeleton, isosurface generation or particle-based flow simulation.

The remainder of this paper is organised as follows. We briefly survey prior work on casting part design, volumetric analysis and foundry defect prediction in section 2. We provide an overview of our software solution in section 3. Section 4 contains an analysis of our implementation, focusing on its performance and limitations. Finally, section 5 presents the conclusions and draws the avenues for future work.

II. Related Work

Despite its importance in industrial development, prevention of errors in casting parts is a relatively neglected study area.

In 1995, R. Yagel was the first to propose the volumetric analysis of casting parts[5]. However, this contribution is mainly focused on the identification of the heavy mass and leaves the task of locating potentially defective areas in the hands of the expert knowledge of the designer.

In a similar vein of research, recent years have witnessed interesting advances in casting process simulation techniques applied to defect detection. More accurately, numerical analysis models had been used to create finite element meshes which can store thermo-physical material properties and various boundary conditions[6], [7]. These models allow for a precise simulation of the molten metal flow and different thermal-stress processes but also require a great amount of computation time and a well defined part model to work with. Unfortunately, those requirements can not always be enforced and only tacit knowledge or trial-and-error may help the designer to achieve a satisfactory result.

III. Visual Defect Prediction

In this section, we describe the process that our algorithm follows to provide the aforementioned defect prediction with a visual tool that helps to improve its flaw forecast.

A. Geometric Analysis

As aforementioned, casting part design is based on CSG to describe the exact geometry of the model. This modelling paradigm, however, is based on complex mathematical structures that require a great amount of processing capability, hindering the usage of current graphics hardware without prior conversion to its Boundary Representation (B-Rep).

This conversion between different geometric representations has been intensively studied in computer graphics literature and recent algorithms allow better approximations to the original model [14], [15], [16]. The resulting polygonal mesh is composed of planar surfaces defined by three vertices that our solution uses to determine its enclosed volume.

Nevertheless, before the triangular mesh can be used as input for our voxelization algorithm, a basic geometric analysis is necessary to ensure its correct processing. This includes surface healing (for better approximation between surface edges), vertex welding (which eliminates any repeated vertex), face alignment (checking the face orientation by recalculating its normal vector) and boundary box determination.

This preprocessing phase significantly reduces the amount of calculations involved in later steps and provides a data set that can be transfered to the graphics hardware for further -and faster- analysis.

B. Voxelization Grid Generation

In other approaches [10], [11], the creation of the voxelization grid is a trivial step that only requires the initialization of a tridimensional matrix of binary values. Due to the nature of our voxelization algorithm and the simulation techniques later explained, a more complex process is required.

Using the bounding box previously calculated and a given accuracy value (expressed in sections of the largest dimension or a fixed measure value), we generate a tridimensional matrix of real values to store the voxelization results. Depending on the complexity of the simulation process it may be necessary to store bigger data structures for each voxel (or even various data matrixes), but it is highly recommended to reduce this information to the minimum, since its three-dimensional nature can increase the memory requirements exponentially and, thus, reduce the performance of the application.

Additionally, in order to optimise the subsequent voxelization process, we create a bidimensional structure for each of the three main planes (XY, XZ and YZ) to store a record of the faces in the model whose projection intersects with each of the cells in the plane. Fig. 1 shows this optimization process being applied to a model that, due to its non-manifold (non solid, topologically incorrect) nature, can not be directly used as input in the aforementioned voxelization algorithms. Please note that, for the sake of clarity, the grid resolution has been greatly reduced in the figure.

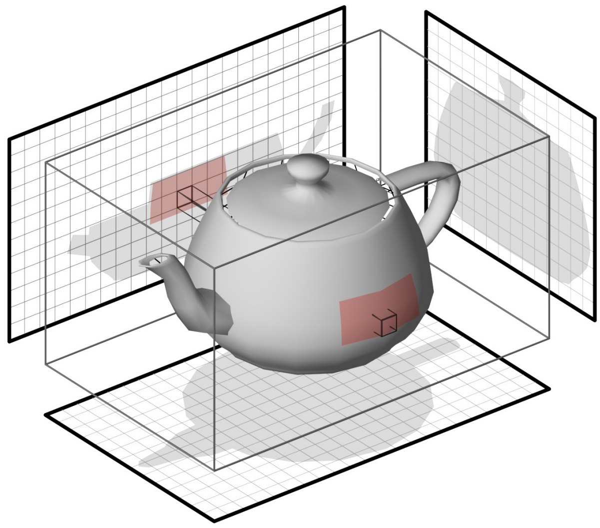

Fig. 1. Voxelization grid optimization.

C. Fault-Tolerant Voxelization

Our voxelization algorithm uses a multi-pass approach to overcome the errors in the casting part model definition. Each pass of our algorithm is performed on a different plane and applies the ray casting method to obtain the volumetric information [17]. This technique involves the testing of the intersection between multiple lines (called *rays*) and the surface of a model and has been widely used in computer graphics for multiple purposes [18], [19], [20].

In our approach, the ray casting process is carried out perpendicularly to the planes of the voxelization grid, generating a ray from the centre of each cell and using the Moller-Trumbore algorithm [21] to obtain the intersection points between the ray and the triangular faces in the model. In case we have at least one intersection point, we examine their location and the normal vector associated with the intersected triangles. In this way, as shown in Fig. 2, we can determine the value for the voxels related to each cell in each plane.

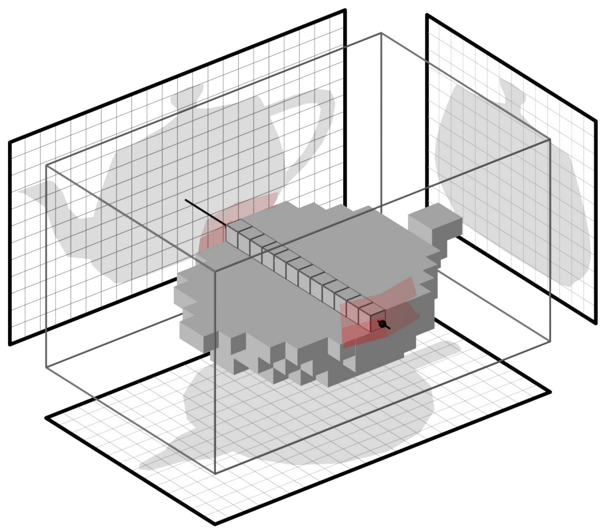

Fig. 2. First pass of the voxelization process (+X axis).

This stage of the process is where most topological errors usually affect the result. Our approach achieves a zero defect rate by introducing a secondary step that analyses the voxel values obtained in each of the three passes and properly adjusts any voxel that present inconsistencies between its associated values.

D. Simulation

Once we have obtained the volumetric representation of the model, it is possible to perform different processes of measurement and simulation to predict defects in casting parts.

Initially, we iterate the positions through the voxelization grid while obtaining different data related to each voxel, such as the distance transformation value or the number of adjacent filled voxels. Based on this data, we can carry out straight line skeletonization [22] and basic solidification analysis that can help to predict hot spots, micro-shrinkages and several mechanical properties.

Further, gathering all the information obtained up to this point, we can simulate fluid flow and solidification with satisfactory accuracy. By using the voxelization results in conjunction with the triangular mesh obtained from the original model, it is possible to analyse the thermal flow and foresee the stress and deformation the casting part will be subject to.

E. Visualization

The correct choice of a visualization method for the information collected in previous steps is an essential issue, allowing the designers to make corrections on the definition of the casting pieces. At this point, the most advanced techniques for volume visualization must serve the designer needs.

Since the current graphical hardware is unable to render volumetric data directly, surface reconstruction methods are needed. These methods redefine the polygonal mesh of the model using the differences between voxel values to create boundaries known as isosufaces. The resulting data allows an easy representation of complex simulations.

Beyond numerical representation, visualization of simulation data overlaps reality layers, highlighting areas that are likely to present defects. It is essential to employ several layers since each layer contains a different data representation of the part model. With appropriate animation and threshold controls, the graphic user interface can provide successful human-computer interaction that empowers the designer to rapidly identify the problem and its solution.

Please note that the visualization process is independent of the voxelization and simulation steps. Typically, voxelization and simulation will be carried out once for each model whereas simulation, based upon data created by both predecessors, as many times as the user requires.

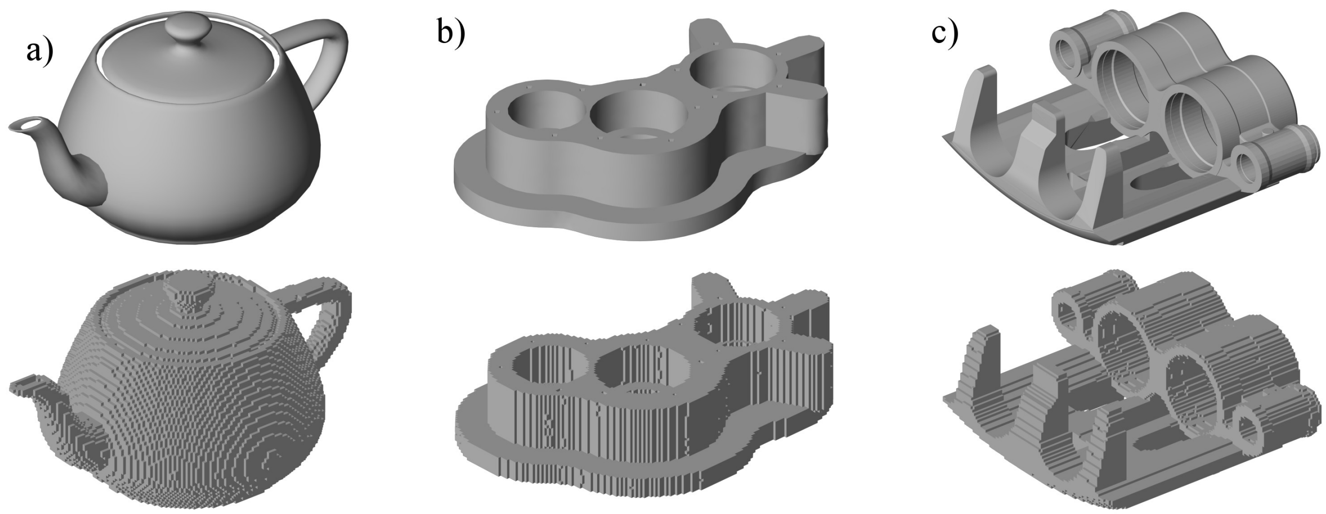

Fig. 3. The Newell Teapot(a), the Pump Splicer(b) and the Break Calliper(c) before and after the voxelization process

IV. Analysis

In this section we evaluate our software solution and discuss the relationship between accuracy and computing time, as well as the limitations of the algorithm.

A. Examples

In order to test the validity of our voxelization algorithm, we have prepared the following three examples, presenting each of one a different topological problem.

- **Newell Teapot:** Defined as the first customizable shape [23], its topology has features which make them difficult to be voxelized by conventional methods. The intersection of the peak and the handle with the central body and the lack of continuity between the body and the lid give the model a hollow semblance.
- **Pump Splicer:** Often used in papers related to computational geometry, this model presents small holes that need to be particularly taken into account during the process of voxelization, which makes it a suitable input to test our algorithm.
- **Break Calliper:** This casting part contains errors in the definition of several of its surfaces. The difficulty of healing such surfaces with methods used in conventional software was the origin of the research whose results are published in this paper.

As shown in Fig. 3, our algorithm overcomes the aforementioned difficulties, generating a accurately voxelized version of each model with just 128 subdivisions in the larger dimension. The results of the voxelization process can be seen in Table I, in which special emphasis is placed on the complexity of the model (a larger number of triangles indicates a more complex topology) and the number of filled voxels (of great importance in subsequent calculations).

TABLE I
Results of the voxelization algorithm.

Model	Triangles	Resolution	Voxels
Newell Teapot	4,096	128x60x63	203,107
Pump Splicer	7,524	128x90x27	109,088
Break Calliper	43,878	128x107x49	187,321

B. Performance

In order to test the performance of the algorithm, we used the break calliper model as an input for it, developing a benchmark on a standard Intel Quad Core i7 920, 2800 with 8GB RAM and using Microsoft Visual C++ 2008. This benchmark focuses on the calculation of the Maximum Inscribed Sphere (MIS) radius to predict any possible micro-shrinkages in the casting part.

Given its regular nature, the voxelization grid must have a large resolution to obtain accurate results. As shown in table 2 the computation time necessary to achieve submillimetric accuracy is far from being negligible. However, since this degree of accuracy is only rarely required and the simulation process is not affected, lower resolutions can be used in interactive applications.

TABLE II
Results of the MIS benchmark.

Resolution	Computation Time	MIS Radius
32x27x13	249 ms	15.0003 mm
64x54x25	796 ms	13.9776 mm
128x107x49	11,154 ms	12.9548 mm
256x214x97	581,525 ms	12.8843 mm

As a general rule, a grid resolution of 256x256x256 is more than enough in almost all cases. For larger resolutions a 64-bit architecture is recommended for better memory management.

C. Limitations

Although our voxelization algorithm has been designed to be as fault-resistant and adaptable as possible, is still has some limitations.

First, our memory requirements may be high due to the use of multidimensional data structures. Moreover, its definition must reserve space in memory to store information in regions without volumetric information which, although not intended to be processed, cannot be ruled out initially.

Second, there is the need to preprocess the geometry of the model to force the step of creating the voxelization grid. For advanced applications requiring a high degree of interaction, it is necessary to handle different sections.

Finally, the possibility of parallel implementation of the algorithm requires taking special care with the writing of the values associated with each voxel of the matrix, as it must be unique.

V. Conclusions and Future Work

Defect prediction in casting parts is a major challenge in foundry-related research and therefore an important area in the development of support tools for the industry. Continuing a previous work, here we deal with three-dimensional visualization of castings to optimize defect prediction. There is a large number of variables involved in the occurrence of such defects but the need to carry out strict quality controls while maintaining a low cost increases the use of these type of applications.

In this paper we have presented a fast and flexible software solution that allows prediction for solidification defects such as hot spots and micro-shrinkages, even for part models with definition errors.

Our volumetric analysis-based solution predicts the apparition of defects in casting parts and suggests improvements during the design phase.In this way, the designer increases his confidence in the validity of a casting part, reducing the number of controls required when the production is finished.

There are several improvements and directions for future work. First, we plan to improve the algorithm implementation by executing certain sections of the code directly on the graphics hardware because of its processing capabilities when handling three dimensions. Second, we will add a method to split the space in order to create a faster and more accurate voxelization. Finally, we are planning to integrate this software into a complete framework.

References

[1] Yoseba K. Penya, Pablo G. Bringas, Argoitz Zabala, "Advanced fault prediction in high-precision foundry production", *6th IEEE International Conference on Industrial Informatics*, pp. 1672–1677, 2008

[2] I. Santos, J. Nieves, Y. K. Penya and P. G. Bringas "Machine-learning-based Mechanical Properties Prediction in Foundry Production" . *Proceedings of the nineteenth annual symposium on Computational geometry*In Proceedings of ICROS-SICE International Joint Conference (ICCAS-SICE), Fukuoka, Japan, 18-21 August 2009, pp. 4536-4541.

[3] J. Nieves, I. Santos, Y. K. Penya, S. Rojas, M. Salazar, and P. G. Bringas, "Mechanical properties prediction in high-precision foundry production.", *7th IEEE International Conference on Industrial Informatics*, 2009

[4] Foffman C. M. "Geometric and Solid Modelling: An Introduction", *Morgan Kaufmann Publishers*, 1989

[5] R. Yagel, S.C. Lu, A.B. Rebello, R.A. Miller, "Volume-Based Reasoning and Visualization of Diecastability", *6th Conference on Visualization*, pp. 359, 1995

[6] Olivier Ludwig, Marco Aloe, Philippe Thevoz, "State Of The Art In Modelling Of Continuous Casting", *6th European Conference on Continuous Casting*, 2008

[7] Jurguen Kaltofen, Ole Koser, Adi Sholapurwalla, Sam Scott, "Casting Simulation Drives Component Development for HPDC", *Die Casting Engineer Journal*, pp. 30–33, May 2009

[8] Shiaofen Fang, Duoduo Liao, "Fast CSG voxelization by frame buffer pixel mapping", *IEEE symposium on Volume visualization*, pp. 43–48, 2000

[9] Gokul Varadhan, Shankar Krishnan, Young J. Kim, Suhas Diggavi, Dinesh Manocha, "Efficient max-norm distance computation and reliable voxelization", 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing

[10] Zhao Dong, Wei Chen, Hujun Bao, Hongxin Zhang, Qunsheng Peng, "A Smart voxelisation Algorithm", 2004

[11] Elmar Eisemann, Xabier Decoret, "Fast Scene Voxelisation and Applications", *International Conference on Computer Graphics and Interactive Techniques*, Sketches, 2006

[12] Duoduo Liao, "GPU-accelerated multi-valued solid voxelization by slice functions in real time", *7th International Conference on Virtual-Reality Continuum and Its Applications in Industry*, pp. 1–6, 2008

[13] Elmar Eisemann, Xavier Decoret, "Single-pass GPU solid voxelization for real-time applications", in *Proceedings of Graphics Interface*, pp. 73–80, 2008

[14] Barton T. Stander, John C. Hart, "Guaranteeing the topology of an implicit surface polygonization for interactive modeling", *International Conference on Computer Graphics and Interactive Techniques*, pp. 279–286, 1997

[15] Gokul Varadhan, Shankar Krishnan, Jersey Liangjun Zhang, Dinesh Manocha, "Reliable implicit surface polygonization using visibility mapping", *4th Eurographics symposium on Geometry processing*, pp. 211–221 ,2006

[16] Marcos de Sales G. Tsuzuki, Fabio K. Takase, Murillo Antonio S. Garcia, Triago de Castro Martins, "Converting CSG models into Meshed B-Rep Models Using Euler Operators and propagation based Marching Cubes", *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, Oct./Dec. 2007

[17] Roth, S.D., "Ray Casting for Modeling Solids", *Computer Graphics and Image Processing*Vol. 18, N. 2, pp. 109–144, 1982

[18] Wei Chen, Wei Hua, HuJun Bao, Peng QunSheng, "Real-time Ray Casting Rendering of Volume Clipping in Medical Visualization" *Journal of Computer Science and Technology*, Vol.18, Issue 6, pp. 804–814, 2003

[19] Stephan Mantler, Stefan Jeschke, "Interactive landscape visualization using GPU ray casting" *4th International Conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pp. 117-126, 2006

[20] Huiyan Jiang, Yudong Zhao, Ning Li, "The Study of 3D Reconstruction Method Based on Dynamic Threshold Method and Improved Ray Casting Algorithm", *Proceedings of the 2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pp. 402–405, 2008

[21] Tomas Moller, Ben Trumbore, "Fast, Minimum Storage Ray/Triangle Intersection", *Journal of Graphic Tools*, pp. 21-28, 1997

[22] Mirela Tanase, Remco C. Veltkamp, "Polygon Decomposition based on the Straight Line Skeleton", *Proceedings of the nineteenth annual symposium on Computational geometry*, pp. 58–67, 2003

[23] Crow Frank, "The Origins of the Teapot", *IEEE Computer Graphics and Applications*, pp. 58–67, 1987