DOE/ID/13362--T8

Simple Visualization Techniques for Die Casting Part and Die Design

Final Report

DOE Project Number DE-FC07-95ID13362

OSURF Project Number 731102

For the period

July 1, 1995 – September 30, 1997

R. Allen Miller,
Professor and Chair, Industrial, Welding and Systems Engineering

Shao-Chiung Lu

Alexander B. Rebello

The Ohio State University

1971 Neil Ave.

Columbus, OH 43210

Phone: (614) 292-7067

Fax: (614) 292-7852

May 1, 1998

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

MASTER

pg
final

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible electronic image products. Images are produced from the best available original document.

ABSTRACT

The objective of this work was to develop and test die casting design evaluation techniques based on the visualization of geometric data that is related to potential defects of problems. Specifically, thickness information is used to provide insight into potential thermal problems in the part and die. Distance from the gate and a special type of animation of the fill pattern is used to provide an assessment of gate, vent and overflow locations.

Techniques have been developed to convert part design information in the form of STL files to a volume-based representation called a voxel model. The use of STL files makes the process CAD system independent. Once in voxel form, methods that were developed in this work are used to identify thick regions in the part, thin regions in the part and/or die, distance from user specified entry locations (gates), and the qualitative depiction of the fill pattern. The methods were tested with a prototype implementation on the UNIX platform. The results of comparisons with numerical simulation and field reported defects were surprisingly good. The fill-related methods were also compared against short-shots and a water analog study using high speed video.

The report contains the results of the testing plus detailed background material on the construction of voxel models, the methods used for displaying results, and the computational geometric reasoning methods used to create die casting-related information from the voxel model for display to the user.

ACKNOWLEDGEMENTS

This work was supported and monitored by the members of the North American Die Casting Association's Computer Modeling Task Group chaired by Bill Walkington of Walkington Engineering. The contributions of the members of the committee were invaluable and the work could not have been completed without this help.

Special thanks are due Walter Smith of DCD Technologies for sharing expertise and providing considerable feedback on the prototype system. Parts and examples were provided by General Motors, Ford Motor Co., and others. Dr. Shuh-Yuan Liou of Ford Motor Co. also provided financial support.

The help of the committee, and particularly the individuals mentioned above, has contributed to the quality of the work. Any errors or inaccuracies, however, are the responsibility of the authors.

Dr. Dave Farson was kind enough to loan the high speed camera used for the water model studies. His generosity made these studies possible. Mary Hartzler and Nils Theller provided considerable assistance and advice in the construction and machining of the dies and the protective steel plates.

The support of the US Department of Energy, the Cast Metals Coalition, and the North American Die Casting Association are greatly appreciated.

TABLE OF CONTENTS

ABSTRACT	I
ACKNOWLEDGEMENTS	III
TABLE OF CONTENTS	V
1. INTRODUCTION.....	1
1.1 RATIONALE.....	1
1.2 OBJECTIVES.....	2
1.3 OUTLINE OF THE REPORT	3
2. BACKGROUND AND APPROACH	5
2.1 RESEARCH APPROACH	5
2.2 GEOMETRY-RELATED DIE CASTING DEFECTS	5
2.3 GEOMETRIC CHARACTERISTICS RELATED TO DEFECTS.....	7
2.3.1 <i>Thermal Mass and Pre-mature Solidification Related Problems</i>	7
2.3.2 <i>Heat Trap Related Problems</i>	8
2.3.3 <i>Characteristics of Fluid Flow</i>	10
2.3 MODELING OF WALL THICKNESS.....	10
2.4 SELECTION OF A PART REPRESENTATION.....	12
2.5 VOXELIZATION.....	13
2.5.1 <i>STL Files</i>	13
2.5.2 <i>Scan Conversion</i>	14
3. GEOMETRIC REASONING METHODS FOR THICKNESS.....	17
3.1 THE EXTRACTION OF THICK SECTIONS WITH DISTANCE TRANSFORM	17
3.1.1 <i>Metrics</i>	17
3.1.2 <i>Qualitative Wall Thickness Evaluation</i>	20
3.1.3 <i>Quantitative Wall Thickness Evaluation</i>	22
3.2 LOCATING THIN SECTIONS OF THE PART AND OF THE DIE WITH SKELETON.....	29
3.2.1 <i>Introduction</i>	29
3.2.2 <i>Thinning vs. Medial Axis Transformation</i>	30
3.2.3 <i>The Generation of 3-D Thinning Skeleton</i>	32
3.2.4 <i>An Alternative Distance-Based Approach</i>	47
3.3 SUMMARY.....	47
4. VISUALIZATION	49
4.1 INTRODUCTION.....	49
4.2 TEMPLATE-BASED RAY CASTING WITH SPACE-LEAPING.....	49
4.3 THREE-LEVEL SCHEME.....	55
4.4 SHADING MODEL.....	58
4.4.1 <i>Previous Shading Models</i>	58
4.4.2 <i>New Contextual Shading</i>	60
4.5 PRESENTATION OF THE ANALYSIS RESULTS.....	67
4.6 RESULTS.....	68
5. VERIFICATION OF THICK/THIN ANALYSIS.....	75
5.1 INTRODUCTION	75
5.2 CASE STUDIES.....	75
5.2.1 <i>Case 1</i>	75
5.2.2 <i>Case 2</i>	80
5.2.3 <i>Case 3</i>	85

5.2.4	Case 4	91
5.2.5	Case 5	96
5.2.6	Case 6	101
5.2.7	Case 7	106
5.2.8	Case 8	115
5.2.9	Re-Design Example	120
5.3	SYSTEM RUNNING TIME	131
5.4	FEEDBACK FROM INDUSTRY	131
6.	QUALITATIVE TECHNIQUES FOR FILL	133
6.1	SPECIFICATION OF GATES	133
6.2	DISTANCE FROM THE GATE	136
6.3	FILL PATTERN	138
6.3.1	Assumptions and Simplifications	138
6.3.2	Description of the method	139
6.4	IMPLEMENTATION	144
7.	VERIFICATION OF FILL ANALYSIS	151
7.1	INTRODUCTION	151
7.2	CASE STUDIES	151
7.2.1	Case 1	151
7.2.2	Case 2	160
7.2.3	Case 3	167
7.2.4	Case 4	172
7.3	WATER MODEL STUDIES FOR MODEL VERIFICATION	176
7.4	SUMMARY AND CONCLUSIONS	196
8.	SUMMARY AND CONCLUSIONS	197
8.1	LIST OF ACCOMPLISHMENTS	197
8.1.1	Design evaluation	197
8.1.2	Thick/thin section	198
8.1.3	Fill visualization	198
8.1.4	Visualization methods	198
	FUTURE WORK	199
	REFERENCES	201

1. INTRODUCTION

1.1 Rationale

Many manufacturability-related factors should be considered along with the functional requirements during the design of die castings if quality parts are to be economically produced. Specifically, a good part design should ensure that the castings will:

- a) fill completely with molten metal,
- b) solidify quickly and uniformly,
- c) eject readily from the die without damage,
- d) involve a minimum of die construction and die maintenance difficulties, and
- e) achieve the part tolerance requirements (NADCA DDC, 1988).

This work addresses a subset of these issues, namely items a), b) and indirectly d) and e).

The objective of this research is to develop die casting design evaluation methods suitable for use during design, particularly early in part design. A die casting visualization system that helps to promote the compatibility of the part design and the die casting process were developed and tested. The system works with virtually every current CAD systems, provides results very quickly, and provides results that are easy to interpret.

The evaluation system is based on qualitative, geometrically-based, reasoning and evaluation techniques. It is the qualitative approach that enables very fast computation times but it also limits the amount of information that can be achieved. The type of questions that we can address are the ones listed above. The type of questions that we cannot answer include:

- What is the temperature at the end of fill?
- How long does the part take to fill?
- What is the solidification time for this part under specified operating conditions?
- What is the riser temperature when the gate freezes?
- How hot is that hot spot?
- What is the velocity of the metal in that thin section?

The point is that, at least to first approximation, answers to the first set of question do not require the level of detail inherent in the second set. The second list is important when the engineer is trying to improve the die design and is developing the process parameters, but a great deal of work can be done with much less detailed information.

The basic approach followed in this work depends on automatically identifying and extracting from CAD data the geometric characteristics of the part that underlie potential fill and solidification problems. Geometric reasoning techniques are used to develop the needed information from the CAD data and the results are graphically presented to the

user. The results of the approach developed in this complement numerical simulation results and when used in combination can produce better casting designs.

The specific tasks addressed include:

- Identification of geometric characteristics that correlate to thermal-related part defects.
- Identification of a part representation to facilitate the extraction of these geometric characteristics.
- Development of geometric reasoning procedures to efficiently extract these geometric characteristics.
- Development of procedures to describe part filling patterns and evaluate potential gate locations.
- Development of visualization techniques to effectively and efficiently display the results.

1.2 Objectives

The general objective of this research is improvement of the compatibility of die casting part designs with the die casting process. Thermal-management, fill paths, the control of process parameters and other die design factors, including cooling line design, length and distribution of spray, and gating system design have a substantial effect on the quality of die castings (NADCA DDC, 1988). Even though control of these factors helps to improve quality, the most economical way to avoid part defects is to design a part that does not have the tendency to create the problems in the first place.

This observation has led us to develop a geometry-based design evaluation approach that requires relatively little computation time and requires relatively little setup on the part of the user. Short-run times and ease of use make the approach particularly suitable for early design when evaluation of many partially developed alternatives is helpful and the opportunity to make changes is at its maximum.

For any given design task, several analysis are usually available. An example can be the tools available for solidification analysis. In this case, designers may simply examine the design visually, apply design rules from a handbook, or run a full FEM simulation. Figure 1.1 depicts the spectrum of techniques. Naturally, a more accurate evaluation demands more computation time, more complete input data, and more sophisticated skills. The suitability of any particular technique depends on the nature of the task itself and the stage in the design process.

As described, this work falls at the geometrically-driven point. Consequently, identification and extraction of geometric characteristics relevant to die castability problems are a major challenge in this research. Reasoning procedures must be able to efficiently detect the presence of these characteristics regardless of the shape complexity. The second major challenge is developing suitable visualization procedures to display the results. The details behind the solutions to these challenges are discussed in the following chapters of the report.

Figure 1.1 The Spectrum of Design Evaluation Techniques

1.3 Outline of the Report

This report provides a detailed presentation of the concepts, techniques and implementation strategy used for the qualitative reasoning and visualization system that has come to be referred to as *CastView*. The material in the report is largely drawn from the Ph.D. dissertations of Shao-Chiung Lu and Alec Rebello, (Lu, 1996; Rebello, 1997). Lu's contributions were primarily in the thickness and thermal evaluation sections. Rebello's work was on the fill related problems. Both contributed to the development and implementation of the visualization procedures and both benefitted from the masters work of (Ouyang 1994) and (Liu 1997) that provided some of the implementation software needed for this work. More details may be found in these documents. Related and supporting information may be found in Lu et. al. (1995, 1996a, 1996b, 1997), Miller and Rebello (1997), Miller et. al. (1997), and Yagel et. al. (1995).

Chapter 2 brief description of some of the typical die casting defects and their relationship to geometric features of the part is presented. This material provides a foundation for the more detailed technical material that follows.

Chapter 3 is a very detailed technical discussion of the geometric reasoning needed to evaluate thickness. Thickness is a simple concept but reasoning about thickness in CAD data is not.

Chapter 4 is a similarly detailed discussion of the innovative methods developed in this work to render and display the evaluation results. It draws very heavily on the digital signal processing and graphics literature. The chapter provides an excellent basis for understanding the technical issues associated with the display of volume data.

Chapter 5 contains several case studies that were performed to verify and validate the qualitative reasoning about thickness. The results of the prototype implementation are compared with numerical simulations. The results all compare very favorably and with a tremendous advantage in computation time.

Chapter 6 contains a description of the concepts and implementation techniques used for fill distance and fill pattern visualization. Chapter 7 contains the results of a number of case studies used to evaluate the qualitative fill approach. The qualitative results were compared favorably with short-shots and numerical simulation. Chapter 7 also provides the details of a water analog study provided additional verification of the approach.

In summary, fast visualization and geometric reasoning methods were developed, implemented as a prototype system, and tested against historical data, simulation data, and in some cases experimental data. The methods work and the results are surprisingly good. The prototype implementation in UNIX is buggy and difficult to use, but with funding from the American Metalcasting Consortium, an improved implementation is under development for the Windows platform. Through this mechanism, the software implementing the results of this work will be distributed by NADCA and will be available for use throughout the industry.

2. BACKGROUND AND APPROACH

2.1 Research Approach

Analysis tools should permit computational testing of a design before significant time, capital and expense are incurred. Analysis should provide a means of determining whether or not the system will work and/or what is required to make it work efficiently (Foreman, 1989).

Much decision-making in design, particularly preliminary design, relies on qualitative or ordinal information, e.g. where is the last region(s) to fill. This is in contrast to a detailed description of the flow vector field in the die. There is therefore a need for evaluation tools that can be used during design for manufacturability analysis that are also robust enough to handle the complex geometries of die castings. A geometry-driven approach meets these needs and avoids the setup, interpretation, and run-time problems associated with numerical simulation.

Qualitative reasoning involves development of methods for building and exercising models of physical systems in spite of incomplete knowledge. Qualitative analysis methods provide a useful complement to analytical and numerical simulation methods by making it possible to derive useful "predictions" about possible behaviors of a mechanism, given incomplete knowledge about its structure and initial state. To reason qualitatively about realistically complex systems, it is necessary to decompose complex systems into simpler mechanisms (Kuipers, 1990). In this work, die castings are decomposed into simple representation using a volume-based scheme called voxels. Thermal and fill characteristics can be reliably evaluated with the voxel representation.

Analysis is performed using the association between various geometric characteristics and die casting defects or other important manufacturing-related criteria. The objective is to help the engineer identify design characteristics that could result in die casting problems. Furthermore, it will help them reduce the search space for good solutions. The tools will help them narrow the number of competing candidates by eliminating unacceptable ones early and easily. The tools will help them quickly and easily step through a number of what-if scenarios allowing them to arrive at designs that are compatible with the die casting process. Because of the shape-independent nature of the mechanisms used, they will be able to efficiently evaluate the designs regardless of geometric complexity.

2.2 Geometry-Related Die Casting Defects

Some of the common defects in die castings are listed below:

- **Cold Flow:** characterized by lines ranging from small lines to deep folds that mark the boundary between different flow paths, frequently accompanied by discoloring. The marks are usually the result of premature solidification of the leading edge of the metal fill pattern. Causes include long flow distance that allows the metal to get cold and poor gating and fill patterns.

- **Non-Fills:** An extreme case of cold flow characterized by "holes" in the resulting casting where the metal did not fill out the cavity. It is often caused by poor fill patterns or cold metal.
- **Heat Pits, Heat Marks, Surface Porosity, Shrink Porosity:** Caused by hot spots in the die. Can be made worse by trapped gases. One of the solutions is to try to increase pressure on the area with the defect by moving the gate closer to the area or by using pressure pins.
- **Heat Sinks, Surface Sinks or Heat Marks:** Causes include incorrect flow path, trapped gas, or cold metal.
- **Leakers:** Causes include cold flow, non-fills, and porosity exposed by machining.
- **Gas Porosity:** Caused by trapped gas which may be improved by changing the metal fill pattern in the cavity to get even, smooth flow (NADCA 1992).

These defects are due to fill and cooling problems. Ultimately the defects are due to the part geometry and placement of the cavity in the die. Ideally, it is best to have directional solidification of molten metal such that the solidification starts from the overflow, advances through the casting, and moves toward the runner. In this way intensification pressure can help pack the metal and feed shrinkage voids formed during solidification (NADCA DDC, 1988).

However, for thin-walled die castings, this ideal solidification pattern is generally not achievable (Herman, 1979) without change to the part geometry. Porosity is difficult to completely eliminate uniform solidification facilitated by uniform part thickness helps to minimize the problems..

Most solidification problems are due to the presence of relatively large thermal masses in the part, pre-mature solidification in the part, and heat traps in the die (NADCA DDC, 1988; Barton, 1981; Herman, 1979). Thermal masses are regions that take relatively long to solidify. Their presence is often accompanied by shrinkage voids in the part, and soldering and oxide deposits on the die cavity surface.

Pre-mature solidification in the part, on the other hand, corresponds to regions that solidify too early. The presence of these solidified regions may prevent feeding other liquid regions. The result can be increased shrinkage voids, incompletely filled regions and woody areas.

Heat traps are regions in the die where heat flow is blocked or slowed. They are relatively hot regions because heat generated in the casting cannot properly flow through those regions to the bulk of the die. The presence of heat traps usually results in shrinkage depressions, stress cracking, heat checking and die maintenance problems. All of these problems are potential causes for increased scrap and manufacturing costs.

Defects caused directly by metal flow include porosity due to air entrainment, air entrapment in the die, and cold shut defects caused by flow paths that are too long. Fill defects manifest themselves in the ways described above and as a general rule, striving

for smooth metal flow will yield better overall castings (NADCA 1992; Ramalingam, Zhao, 1987).

Past experience is the principal basis for design of die castings and die casting dies. However, geometric parameters can be used to improve current industrial practice with respect to filling (Ramalingam, Zhao, 1987) and solidification.

In the next section, geometric characteristics drawn from the literature and/or used by experienced die casters to prevent defects will be identified. The correlation of these geometric characteristics with manufacturing problems and an efficient approach to extract the geometric data from a voxel model will be discussed

2.3 Geometric Characteristics Related to Defects

2.3.1 Thermal Mass and Pre-mature Solidification Related Problems

The production rate, among other things, depends on the "thickness" of a casting (Barton, 1981). In general, wall thickness is not directly responsible for meeting the functional requirements including strength. The skin layer of a casting, formed at the initial contact where the metal splashes against the die surface, has stronger mechanical properties than the inner bulk portions of the casting. This is due to its finer grain structure and lower porosity. From this perspective, in cases where a thin wall cannot provide adequate strength, it is better to use strengthening ribs rather than increasing the wall thickness (NADCA DDC, 1988; Barton, 1981). On the other hand, if the wall is too thin, metal may not have enough time to completely fill the cavity before it freezes. Therefore, the wall thickness of a part should be designed so that it promotes good metal flow, but at the same time does not unnecessarily delay the solidification.

From a qualitative point of view, wall thickness should be kept as uniform as possible. Abrupt changes in wall thickness may result in isolated thermal masses and pre-mature solidification that can cause casting defects such as shrinkage voids. This scenario is demonstrated in Figure 2.1 (Herman, 1979).

Pre-mature solidification is responsible for many die casting defects, including incomplete fill and woody areas. A thin section of a part contains a relatively small amount of heat and hence may solidify earlier than surrounding regions. The pre-mature freezing may block the metal being pushed into the rest of the casting to fill it completely or to compensate for shrinkage. The presence of non-uniformly thin sections can also cause a perturbation in the metal flow and thus increase the amount of turbulence and entrapped gas. From the perspective of castability, relatively thin sections can cause as many difficulties as the isolated heavy sections. They both significantly complicate die design and process control.

There may be a quantitative range of preferable wall thickness for different die casting alloys, depending on the size of the machine a company possesses. Within this range, a part can be manufactured most economically with standard die casting practice. Thick parts usually incur a longer cycle time. A casting can be regarded as too thick when its cycle time is increased to the point where die casting is no longer the most economical manufacturing process. Conversely, a large machine is usually needed to die cast a thin

part. The die casting machines that a company possesses may fall short of providing the required filling rate and metal pressure when a part is too thin. There are no hard and fast rules governing the maximum and minimum wall thickness of a diecasting, and this quantitative range is only a rough guideline. Therefore, there is also a need for diecastability programs to provide quantitative calculation of the wall thickness of a part.

Figure 2.1 Example Showing the Formation of a Shrinkage Void in a Heavy Section of a Casting (Herman, 1979)

2.3.2 Heat Trap Related Problems

The most obvious cause of heat traps is the existence of relatively confined regions in the die which creates excessive restriction to heat flow (Herman, 1979). In these thin die regions, heat flow paths converge so heat generated in the casting is difficult to properly flow through them. Thus, they are usually subject to high temperatures, causing shrinkage depressions, stress cracking, heat checking and die maintenance-related

problems. There are several design rules, e.g. the distance between ribs, the radius of holes, and the dimensions and locations of slots, which address this problem.

Rather than compliance with those design rules, the real issue is that there must be enough die steel for heat removal. From a geometric point of view, insufficient die steel is associated with geometric characteristics such as narrow holes in the part, i.e. thin die regions. For diecastings with heat flow constraints, die casters usually have to try-out sophisticated thermal control mechanisms to balance the heat flow. Elimination of this problem, where possible, often reduces die wear, improves part quality, reduces die maintenance costs, etc. The information on the thin die regions can also facilitate planning the cavity placement. In general, cooling problems can be reduced if the relatively thin and confined regions can be oriented such that they face more open areas in the die. Hence, identification of regions with insufficient die steel from the perspective of a part design, i.e. the thin cavities in a part, will allow designers to consider alternatives that have fewer heat dissipation problems, or at least will help them recognize the existence of these problems.

An abstract description of the method for dealing with the thermal- related problems, and the associated geometric characteristics with these problems is summarized in Figure 2.2.

Figure 2.2 Methodology For Dealing with Solidification-Related Problems

2.3.3 Characteristics of Fluid Flow

Fluid flow during the filling of a die casting is dominated by inertia and is in the turbulent flow regime or beyond with Reynolds numbers in excess of 2000 due to flow of metal through the quickly changing-geometric cross-sections of the gate and casting cavity. Velocities are as high as 600-1800 ips. Typically, the cavity is filled with molten metal in 0.010 to 0.060 seconds. There is much jetting, splashing and atomization of the molten metal as the cavity fills (NADCA 1996a; Stuhrike and Wallace, 1965; NADCA 1996b; Smith and Wallace, 1963).

The fluid stream, for a straight gate, flows through the cavity without much change in its shape or direction until it impinges upon a wall. The stream impinging on a wall splits approximately following the cavity perimeter in the vicinity of the impingement. This inertia-dominated flow distinguishes die casting from sand casting and slower filling processes (Venkatesan and Shivpuri, 1995).

A jet of liquid metal entering the cavity retains its original direction and cross section as determined by the smallest cross section of the gate. The stream flows without change in shape or direction through the cavity till it impinges a wall. The jet strikes the wall of the cavity in a series of pulsations. After a certain amount of metal has collected at the far end of the cavity, most of the injected fluid is absorbed by the pool of metal that fills the whole cross section of the cavity, although some metal still flows along the cavity walls (Smith and Wallace, 1963). As the metal flow strikes the walls of the cavity, it changes direction, distributing the flow volume and hence reducing the volumetric flow rate.

The planning of flow paths is a critical step since that is where the designer establishes the philosophy of the gating system. When molten metal must travel a long distance through the die, it is usually best to carry it as far as possible through a runner and then gate it into the cavity. The flow distance through the cavity should be minimized. In addition, all parts of the cavity should become filled at the same instant (NADCA 1996b). Prior to the use of simulation, the only means to identify the shortest flow distance was to have the metal flow along the direction of the shortest dimension of the cavity.

These characteristics can be accounted for with a qualitative model that utilizes a volumetric representation of the die casting. The geometry of the casting itself is used to describe the fill pattern and flow distances.

2.3 Modeling of Wall Thickness

The thermally-related geometric characteristics identified in previous sections are based on the concept of thickness. For a flat plate, or shell objects, thickness is easy to define, but this is not true for regular die castings. Figure 2.3 shows the cross section of a simple part composed of a flat plate and an off-center circular hole. It is hard to decide which thickness should be used in the evaluation.

In practice, experienced die casters visually check the wall thickness on engineering drawings, perhaps with the help of a scale. To facilitate the wall thickness evaluation, some commercial CAD systems, e.g. Pro/ENGINEER, provide semi-automatic thickness evaluation tools (Pro/ENGINEER User's Manual). However, it requires users to slice

through the objects manually to define cross sections. Since the measurement of wall thickness is performed in 2-D, the results are very sensitive to the inclination of the cutting planes.

Figure 2.3 Difficulties in Defining the Wall Thickness

To systemize the measurement of wall thickness, Cocks (1983) suggested the use of the diameters of maximum inscribed spheres as a representation of wall thickness for diecastability evaluation. The same idea has been adopted in applications for injection molding (Unruh, 1992; Beiter and Ishii, 1991; Menges et al., 1988). Basically, it requires the calculation of many inscribed spheres that just touch the surface of a die cavity, and their diameters are then used to represent the wall thickness. Based on these spheres, local mass concentration can be identified by comparing their diameters, as shown in Figure 2.4. This approach provides die casting engineers with a systematic measurement of thickness, and it can be easily extended to cover the extraction of thin die regions. However, it is very cumbersome to use manually.

Even with the help of CAD systems, the calculation of inscribed spheres is a tough task for a polygon model, not to mention the various surface definitions in a regular B-Rep model. In this research, the voxel model provides us with a discrete number of points from which the inscribed spheres can be computed easily through the use of the distance transformation and thinning. The details are discussed in chapter 3.

Figure 2.4 The Identification of Thick Sections with Inscribed Spheres

2.4 Selection of a Part Representation

One of the most critical issues in this research is the identification of an appropriate representational scheme to support calculations of the type described above. Conventional CAD representations provide a reasonably robust approach in terms of geometric modeling and manipulation. However, it is quite difficult to derive geometric characteristics such as wall thickness or fill distance from these models.

In the voxel representation, an object is decomposed into a collection of identical cubic units, called voxels (Jense, 1989). The digitized object is then stored in a large 3-D frame buffer (array) where each voxel contains a value to indicate its occupancy state. In other words, each voxel is associated with a value indicating if it belongs to the die (with the value 0) or to the part (with the largest possible integer value). Other values may be assigned to the voxel as well depending on the application. Figure 2.5 shows a 2-D example.

Voxel models were first used in applications where the data sets themselves are voxel-based, e.g. computed tomography (CT) in biomedical applications, and the Frame Buffer stores the uniformly sampled density. Because the large storage space is required, applications were somewhat restricted. However, with the increasing maturity of computer technology, the memory requirement becomes a less serious issue, and the voxel-based models are now used in areas such as 3-D image animation and simulation (Ebert and Parent, 1990).

CAD Model

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	0	0	0	0	0	*	*	*	*	*	0	0	0	0	0
0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0
0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0
0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0
0	*	*	*	*	*	*	*	*	*	*	*	*	*	*	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Voxel Model

Figure 2.5 An Example of The Voxel Representation in 2-D

The major concern of using voxel-based representations in CAD-related applications is the coarse quantization of complex surfaces. The relatively low accuracy of voxel models in comparison with the precise mathematical descriptions of the surfaces in conventional CAD models might cause problems in applications such as tool path and inspection plan generation, but is not an issue in qualitative applications.

Since evaluation is performed on the collection of uniform cells, the reasoning algorithms are not sensitive to the shape complexity. The uniformity of voxel representations allows modeling and reasoning to be easily controlled. Lastly, the generation of the voxel model is nearly automatic and much easier than the complex meshes used in numerical tools.

2.5 Voxelization

2.5.1 STL Files

Since the voxel model is not directly supported in most CAD systems, a mechanism to convert a CAD model to a voxel model is required. To make the system platform independent, the voxelization program cannot be built upon the data structure of any particular CAD system. Practically all CAD systems are capable of generating tessellated models from their representations. The polygon models that have resulted from this tessellation are used for various applications and primarily for stereolithography. SLA and STL files are the most common type of polygon file and can be generated by practically every CAD system. The STL model in this case is a collection of triangles that cover the surface of the part and provide a simple but approximate representation of the part surfaces. The file consists of the vertices and the outward pointing normal vector for each triangle.

These models are a polygon equivalent of a B-rep. If the tessellation is initiated using a solid model, the resulting SLA file should be closed spatially. However, there are no space closure conditions that must be satisfied which means the resulting model may, due to various errors, not form a solid.

The strategy used in this work is to use the polygon-based (i.e. facet) model, as a vehicle to convert from the different B-Rep models to the voxel-based model (see Figure 2.6).

Figure 2.6 Depiction of transformation from CAD to voxel model

2.5.2 Scan Conversion

The system must automatically generate the voxel model from the SLA file. This process is called scan-conversion.

Kaufman and Shimony (1986) discuss algorithms for scan-converting 3-D geometric objects into their discrete voxel representations within a large Cubic Frame Buffer (CFB) of unit cells called volume elements or voxels. These include 3-D lines, polygons, and polyhedra. These algorithms do the scan conversion with linear computational complexity in the number of voxels written to the CFB and yield a cellular tessellation of the original volume. The algorithm takes advantage of plane and edge coherence to reduce conversion time.

Besides scan converting the interior of the part, the boundary of the part, i.e. the faces, must also be scan converted to improve surface smoothness and fidelity. The objective is to yield the longest possible scan lines so that the error is distributed over a longer line segment while reducing the undulations between scan lines.

The first task in scan converting a polygon is to find the envelope of the polygon along the X, Y, and Z axes as illustrated in Figure 2.7. The smallest box is chosen as the minor axis W. The largest is chosen as major axis U. When two dimensions are equal, either of is chosen as the W axis and the third axis is chosen as the U axis. The remaining axis is specified as the V axis.

Beginning with the minimum V value for the polygon, the intersections of a series of planes parallel to the UW plane and the edges of the polygon are computed as shown in Figure 2.8. These yield the U and W values of the intersections. A 3-D line joining each pair of endpoints is then drawn. The set of lines forms the polygon surface.

The best known line algorithm, Bresenham's Digital Differential Analyzer algorithm, is optimal and only utilizes integer arithmetic to scan-convert 2-D straight line segments. Kaufman and Shimony (1986) present a 3-D line algorithm that is actually a double Bresenham's 2-D line algorithm. The algorithm uses only integer arithmetic, and only addition, subtraction, shifting, and testing. Cohen and Kaufman (1991) present line algorithms categorized by connectivity. These include 3-D algorithms used for voxelizing 6-, 18-, and 26-connected lines. The 6-connected line algorithm is used for the generation of the surface of the polygons.

In 3-space two lattice points are said to be 26-adjacent if they are distinct and each coordinate of one differs from the corresponding coordinate of the other by at most 1. Two lattice points are 18-adjacent if they are 26-adjacent and at most two of their coordinates differ. Lattice points are 6-adjacent if they are 26-adjacent and differ in at most one coordinate (Kong, Rosenfeld, 1989). If viewed in terms of volumetric unit cubes, two voxels are 26-adjacent if they share a vertex, an edge or a face, they are 18-adjacent if they share an edge or a face, and are 6-adjacent if they share a face (Cohen, Kaufman, 1991). In this case, the voxel for the point (x, y, z) is defined to be the continuous region (u, v, w) such that $x-0.5 < u \leq x+0.5$, $y-0.5 < v \leq y+0.5$, and $z-0.5 < w \leq z+0.5$ (Kaufman, 1987). 6-, 18- and 26-connected lines are the discrete representation of lines along which consecutive voxels belonging to the lines are 6-, 18- and 26-adjacent respectively.

Because the major and minor axes for each polygon are chosen based only on the dimensions of the polygon, artifacts such as "hanging" voxels may be created at the edges connecting some polygons. This is due to the fact that scan lines between the adjacent polygons at the adjoining edges are not identical and because of floating point versus integer arithmetic. The intersections are computed using floating point arithmetic but the actual scan line voxels are computed using integer arithmetic.

Figure 2.7 Dimensions of the Bounding Box of the Polygon is Used To Determine the Major and Minor Axes

Figure 2.8 Polygon Scanlines Drawn Between Intersection Points of Polygon and Plane With Incremental V Axis Values

3. GEOMETRIC REASONING METHODS FOR THICKNESS

The analysis techniques developed to extract information from the voxel representation are presented in this section. This material is quite mathematical and may be skipped by those interested only in applications.

3.1 *The Extraction of Thick Sections with Distance Transform*

The distance transformation is a tool used in image processing. Basically, it computes in each voxel of an object the shortest distance to the so called "designated" voxels. In the case of wall thickness calculation, the "designated" voxels are specified to be those voxels on the boundary. In this case, the distance transform represents the radius of the inscribed spheres, i.e. the shortest distance to the part boundary. Based on this transform, heavy sections can be identified by extracting voxels with relatively large distance values because they are farther away from the part boundary, as illustrated in Figure 3.1. In the following, we discuss how the distance transform is defined and how it can be computed efficiently.

3.1.1 Metrics

Distance is defined in terms of metrics. Euclidean distance is commonly used in continuous space and the every day world. However, the computation of this "true" distance is in fact a global operation. Even in voxel models, computing the Euclidean distance from a voxel to its closest "designated" voxel(s) requires knowledge of the whole image; and thus is computationally prohibitive. To efficiently obtain a DT, many other metrics have been introduced in the discrete world to approximate the Euclidean distance (e.g. Rosenfeld and Pfaltz, 1968; Borgefors, 1984). The computation of distance based on these discrete metrics is performed locally in a small neighborhood, a limited number of voxels at a time. In principle, the discrete distance between two points is calculated by accumulating the local distance along the shortest discrete path between them. Consider two points p and q in the space Z defined by a uniform grid. A discrete path S_k is a series of distinct points $p=p_1, p_2, \dots, p_n=q$ in Z such that p_i and p_{i-1} are connected by one of the prime vectors v_i defined for a given metric (Verwer, 1991). Each prime vector is associated with a length, i.e. the local distance, which when summed along the path constitutes the total length of the path $L(S_k)$. That is

$$S_k = \{w_1, w_2, \dots, w_n\}, \quad w_i \in \{v_j\}$$

$$L(S_k) = \sum_{j=1}^n |w_j|$$

The distance between two points p and q is then the minimum length of the all possible paths between them, i.e.

$$D(p, q) = \text{Min}\{L(S_k)\} \quad \forall S_k \in S$$

where S is the set of all possible paths between p and q

Figure 3.1 Identification of Thick Sections with Distance Transform

Figure 3.2 shows 2-D examples of two classes of metrics defined in 3x3 and 5x5 neighborhoods. In the case of 3x3 neighborhoods, there are two types of prime vectors, while in 5x5 neighborhoods there are 3 types of prime vectors (as shown in Figure 3.2). These prime vectors are not necessarily linearly independent, but they are not integer multiples of any other prime vector.

By varying the local processing window and the length of the prime vectors, we can define many different metrics. For example, for a 3x3 case we can at least define 3 types of metrics: city block ($a=1$ $b=2$ or ∞), chess board ($a=1$ $b=1$), and 3-4 chamfer metric ($a=3$ $b=4$). If defined this way, the general chamfer distance between two discrete points $p(x_1, y_1)$ and $q(x_2, y_2)$ is:

$$D(p, q) = b * m + a * n$$

where $m = \min(|x_1 - x_2|, |y_1 - y_2|)$ and $n = \max(|x_1 - x_2|, |y_1 - y_2|)$

When $a=1$ and $b=1$, we obtain the chess board distance, i.e.

$$D(p, q) = \max(|x_1 - x_2|, |y_1 - y_2|)$$

On the other hand, for the city block metric, no diagonal steps will be made, so distance defined in this metric is simply $D(p, q) = |x_1 - x_2| + |y_1 - y_2|$

In general, more accurate distance can be obtained by expanding the processing window. In the extreme case, if the processing window is expanded to cover the whole object, a true Euclidean distance can be obtained. Figure 3.3 shows an interesting comparison among the discrete spheres with the 3-D versions of the previously described 2-D metrics (Borgefors, 1984).

Figure 3.2 2-D Examples of Two Classes of Metrics Defined in 3x3 and 5x5 Neighborhoods

Figure 3.3 Various Discrete Spheres

3.1.2 Qualitative Wall Thickness Evaluation

In applying the DT in qualitative wall thickness evaluation, i.e. the identification of local mass concentrations, either the city-block or chess-board metric can be used. It can be seen that both of them are valid metrics, i.e. (Porter 1966),

- a) $D_{cb}(P_1, P_2) \geq 0 \quad \forall P_1, P_2 \in Z$
- b) $D_{cb}(P_1, P_2) = 0 \Leftrightarrow P_1 = P_2$
- c) $D_{cb}(P_1, P_2) = D_{cb}(P_2, P_1)$
- d) $D_{cb}(P_1, P_2) + D_{cb}(P_2, P_3) \geq D_{cb}(P_1, P_3)$

However, heat can only flow across the face-connected voxels according to the Fourier's law. Thus, the city-block metric is more suited to extract the heavy sections which usually accompany thermal-related part defects, and most importantly it is extremely efficient in terms of computation. Therefore, the city-block metric is used here to extract the heavy mass region in the part.

Borgefors (1984) proposed an efficient 2-pass sequential algorithm for distance transformation with all the chamfer metrics, including the city-block and the chess-board metrics. First of all, let us explain the algorithm in the one-dimensional case. Figure 3.4 shows a 1-D voxel model, initially with the value of 0 for the boundary voxels and the largest possible distance (indicated by *) for the "inside" voxels for which we want to calculate their closest distance to the boundary. There are two steps in this algorithm, and each step is associated with a graphic mask shown in Figure 3.5. In the first step, the forward mask is initially placed on the voxel model with the "0" covering the second voxel to the left. The constants on the mask are then added to the values of the voxels it covers, and the minimum value is assigned to the second voxel. The forward mask continues moving to the right and updating the voxel values until it reaches the end. After the first traversal of the data, each voxel has the distance to its nearest boundary to the left (see Figure 3.5). The same process is repeated for the second pass with the backward mask, but moving the mask to the left as indicated in Figure 3.5. This step first computes the minimum distance for each voxel to the right and then compares that with the previous value, i.e. minimum distance to the left. Thus, the results of the two complete passes generate the correct distance for each voxel to its nearest boundary voxel(s) (see Figure 3.4).

The same principle can be easily extended to 3-D distance transformations. In this case, two 3-D masks with the values shown in Figure 3.6 (for the city-block metric) traverse through the volume (Borgefors, 1984). The forward mask travels from top to bottom, right to left, and front to back, while the backward mask travels in the opposite direction. As in the one-dimensional case, the value (local distance) in each mask is added to the value of the voxel it covers while moving the masks over the volume. The minimum value of these sums is then assigned to the "zero voxel" (the voxel covered by the mask-value 0). If the distance transformation is applied on the voxels of the part relative to the die, all the voxels which belong to the die have the initial value 0 while the rest of the

voxels (i.e. the part) have the initial value greater than the maximum possible distance, e.g. 255. Thus, two complete passes of this volume, once with the forward mask and once with the backward mask, will generate the distance from each voxel in the part to its closest voxel(s) in the die. Figure 3.7 shows how this algorithm works in a simple 2-D cross section with the city-block metric, and the relatively heavy mass region is also marked.

```

0 0 * * * * 0 * * * 0 0
0 0 1 2 3 4 0 1 2 3 0 0
0 0 1 2 2 1 0 1 2 1 0 0

```

Figure 3.4 Example Illustrating the Distance Transformation in 1-D

Figure 3.5 The Masks for 1-D Distance Transformation

Figure 3.6 The 3-D Masks in Distance Transformation with the City-Block Metric

3.1.3 Quantitative Wall Thickness Evaluation

In addition to the identification of relatively thick sections, sometimes die casting engineers want to quantitatively evaluate the wall thickness of a design. However, due to the characteristics and limitations of voxel models and digital distance transformations, it is quite difficult for the voxel-based approaches to compute the exact wall thickness. Nevertheless, for many practical applications, a rough approximation of the wall thickness is adequate, as long as the limitations of accuracy is known. In this section, a procedure which can quantitatively identify the thick section of a part is presented.

The first task in the quantitative evaluation is to identify a metric that can better approximate the Euclidean distance. As shown in Figure 3.8, both the city-block and the chess-board metrics described in the previous section fall short of this. This is because the local distances selected for these metrics are far from optimum values. The first (the city-block metric) always over-estimates the Euclidean distance, while the other (the chess board) tends to under-estimate it. For example, when $m=n$ (two points P and Q form a line at 45 degrees to the grid lines) the relative error for the city block metric is $(2-\sqrt{2})/\sqrt{2}$, i.e. more than 40%. In the same case, the relative error for the chess board metric is $(\sqrt{2}-1)/\sqrt{2}$, still about 30%. Therefore, a different metric is needed. As for the local processing window of the metric, computational efficiency is a major concern here. Thus, even though the relative distance error can be kept arbitrarily small by utilizing a larger processing window, the metric we use is confined in the 3x3x3 neighborhood.

The 2-D masks

Original part

Distance value
after the
forward pass

The minimum
distance with
two complete
passes

Figure 3.7 Example Illustrating the Distance Transformation in 2-D

The distance between P and Q in:

City Block Metric = $m+n$

Chess Board Metric = m (for $m > n$)

$(1, \sqrt{2})$ Chamfer Metric = $\sqrt{2}n + m - n$

Euclidean = $\sqrt{m^2 + n^2}$

where m and n are horizontal and vertical steps between P and Q ($m > n$)

$$m < \sqrt{m^2 + n^2} < \sqrt{2}n + m - n < m + n$$

Figure 3.8 Comparison of the Various 2-D Metrics

Significant research has been dedicated to compute a more accurate DT. Algorithms have been published for the so called Euclidean DT, which contains the exact distance between grid points (Danielsson, 1980; Borgefors 1984). The basic principle of this approach is to keep track of both the "X" and "Y" increments along the shortest path, and the exact distance between grid points is obtained by computing the length of the vector, i.e. (X,Y). Even though it is extremely accurate, the limitations of this approach are the memory requirements and the computational efficiency. Four passes through the whole object are needed as opposed to traversing the volume only two times in most other metrics. At least two integer values are stored for each voxel, and the computations of the vector length during processing is very time-consuming (Borgefors, 1984).

For a more accurate distance measure, integer-based chamfer metrics are usually used along with the corresponding scale factor. This type of metric has the advantage of being both computationally efficient and accurate. The integer value is chosen by scaling the theoretical optimum local distance to whole numbers, and a scalar is then used to obtain the calculated distance (Verwer, 1991; Borgefors, 1986). The choice of the local distances for a chamfer metric is not arbitrary, the selected distances have to respect the triangular rule in addition to being symmetric and positive-definite (Verwer et al., 1989). In other words, valid local distances, i.e. a and b in Figure 3.2, should be selected such that :

$$a \leq b \leq 2a \quad (3.1)$$

This will ensure that a diagonal move is not shorter than a vertical/horizontal move, and not longer than two vertical/horizontal moves so that diagonal moves will be neglected in the shortest path. For 2-D local distances, a natural choice would be to change the diagonal distance to $\sqrt{2}$ while keeping the vertical/horizontal distance as 1. However,

these intuitively appealing local distances are not the optimum choice, even though they are much better than original local distance pairs, i.e. (1,2) in the city-block or (1,1) in the chess-board. In this metric, the error between the estimated distance and the Euclidean distance is $\sqrt{2}n + m - n - \sqrt{m^2 + n^2}$ (see Figure 3.8). This error is minimized when $n=0$ (points P and Q are along the same horizontal grid line) or $m=n$ (points P and Q lie on a line 45 degree to the grid lines). The maximum error occurs when the first derivative with respect to n is zero, for any given m , i.e. $n = (1/\sqrt{2+2\sqrt{2}})m$. In this case, P and Q make a 23.2 degree angle to the horizontal grid line, and the maximum relative error is about 8%, i.e. $0.09m/\sqrt{m^2 + n^2}$.

Borgefors (1986) has presented the optimal local distances for chamfer metrics up to 7X7 by minimizing the maximum deviation of the chamfer distance from the Euclidean distance. The optimization is calculated based on the distance from the origin to a line $x=\text{constant}$. For 3x3 chamfer metrics, the optimum vertical/horizontal distance is 0.95509, and the diagonal one is 1.36930 (Borgefors, 1986). In this case, the maximum relative error is 3.2%. Since integer values are preferred in terms of memory usage to floating point values, he also presented several integer combinations to approximate the calculated optimum local distances. These integer approximations are obtained by multiplying the optimum floating point local distances by an integer factor N , and then rounding them off to the closest integers. With the integer approximations, the computation is more efficient and the actual distance can still be obtained at the end of the transformation by multiplying back the scalar factor, i.e. N . The maximum errors with the integer local integer distances can be found in Borgefors (1986). Vossepoel (1988) pointed out that the accuracy of the integer-based chamfer distance can be further improved by using a non-integer re-scaling factor. The optimum factors were then published by Borgefors (1991).

Most of the work done regarding chamfer metrics concentrated on 2-D distance transformations. For 3x3x3 chamfer metrics, another type of prime vector along the diametric direction is added. To ensure that the newly introduced distance is used properly, these local distances have to be selected according to:

$$a \leq b \leq c, b \leq 2a, \text{ and } c \leq b+c \quad (3.2)$$

Verwer (1991) presented the optimum local distances and the accompanied maximum relative errors based on the distance from the origin to a unit sphere. The results of his work are shown in Table 3.1. From this table, it can be seen that even with integer approximation on a small processing window (3X3X3), the relative error of the chamfer distance can be easily kept under 10%.

a	b	c	Scale Factor	E-max %
1	1	1	0.7887	26.79
1	2	3	1.3660	26.79
2	3	4	2.2247	10.1
3	4	5	3.0725	7.94
4	6	7	3.2913	6.79
7	10	12	7.4011	6.39
19	27	33	20.2355	6.11
1	$\sqrt{2}$	$\sqrt{3}$	1.0641	6.02

Table 3.1 Some Integer Local Distances and Their Associated Scale Factors

This value only indicates the distance errors between grid points based on a voxelized object. In reality, those "grid" distances may not represent the actual wall thickness due to "resolution errors". Specifically, the errors can be classified into two types. The first type of error comes from the fact that the distance is only computed at a fixed number of sampling points. However, the actual maximum inscribed spheres in continuous space may not be centered at voxel grid points. Thus, the local maximum values in the DT may not truthfully represent the radii of the maximum inscribed spheres, i.e. half of the wall thickness.

Figure 3.9 illustrates this type of error in 2-D. In 3-D, the actual wall thickness can be greater than that indicated by the discrete DT by $\sqrt{3}$ times the voxel size, though it will never be smaller than the corresponding discrete distance in the DT.

The second type of error is due to the rough approximation of the object boundary by binary voxels in the voxelization stage. In other words, it does not account for the real positions of the surfaces within each boundary voxel. Of course, the faceted model also introduces some errors to the wall thickness calculation when approximating the surface of an object by polygons. However, this error is relatively small in comparison with the resolution error, and it can be adaptively refined to an acceptable level. On the other hand, the upper bound of the distance error resulted from the further approximation of a faceted model by binary voxels can be as large as $\sqrt{3}$ times voxel size. This is also illustrated in Figure 3.9. In summary, the actual wall thickness can be as much as $2\sqrt{3}$ times the voxel size larger than the one indicated in DT, or up to $\sqrt{3}$ times the voxel size smaller than the DT-based wall thickness.

Figure 3.9 Errors in Quantitative Wall Thickness Evaluation Caused by Voxel Resolution

Due to the limitations on computer memory, it is impractical to arrive at the wall thickness with an arbitrarily high precision in DT by just utilizing a finer grid. Since 2-3 bytes of memory are required for each voxel in thermal-related evaluation alone, the total memory demand will be at least 54 M-Bytes for a 300^3 volume. For situations where a more accurate measure is required, two methods are used. First, a non-cubic volume is utilized rather than the commonly used cubic one. For some die castings, the geometry is not "box-shaped", i.e. the part may be elongated in one direction but thin/narrow in the others (e.g. Figure 3.10). Therefore, representing such an elongated object with a cubic volume is very inefficient because most of the voxels are empty. To use memory more efficiently, we need to adjust the shape of the volume dynamically so that the volume can better fit the object. Secondly, we can first use a coarse resolution volume as a filter to find regions of interest. If more detailed analysis is needed, the user can always use the CAD system and physically cut the part to retain just these regions. Then a much finer resolution can be used for these localized regions to obtain a more accurate measurement.

Figure 3.10 Example Part that Would Use Memory Inefficiently with a Cubic Frame Buffer

3.2 Locating Thin Sections of the Part and of the Die with Skeleton

3.2.1 Introduction

The DT-based reasoning procedures described above can efficiently extract the heavy mass regions of a part. However, the same strategy cannot be directly used for the extraction of thin regions; otherwise all the voxels on the boundary would be selected as thin. This is because wall thickness can be truthfully represented only by the distance transform values of the voxels that lie near the medial surface. In other words, only those "medial" voxels, not all voxels, which have relatively small inscribed spheres are thin regions. For example, the dotted circles in Figure 3.11 are not centered at the medial axis, and therefore are not thin regions. To identify the thin regions, therefore, the voxel model needs to be enhanced by labeling the voxels of the medial surface. In addition, for the extraction of the geometric characteristics of the die, the distance transformation and the identification of the medial surface should be applied to the die. Based on this representation, thin regions of the part and of the die can be extracted by identifying the medial surface voxels associated with smaller spheres.

Figure 3.11 Identification of Thin Regions.

Extension of the 2-pass distance transformation algorithm (Borgefors, 1984) to cover the die is rather straightforward. To accomplish this, all the voxels on the part boundary need to be identified. This can be done at the voxelization stage when the boundary is scan-converted. These "boundary" voxels are assigned the value 0, and all the other voxels are initialized to the largest possible distance value. The boundary voxels are then used as the starting points from which distances are measured. While computing the distance transformation, we use the last bit of the voxel value to distinguish the voxels of the part from ones of the die (odd vs. even). In this way, after 2 traversals of the volume, voxels of both die and part will have a distance value indicating their shortest distance to the boundary.

In contrast to the previous task, generation of an appropriate medial surface representation, the so called skeleton, is not an easy task. Especially for the shape complexity of diecastings, many issues are involved to develop a robust algorithm. In the following sections, we will discuss this topic in detail.

3.2.2 Thinning vs. Medial Axis Transformation

The skeleton is a compact representation of a geometric object. It was used primarily in the field of image processing and pattern recognition because it is an abstract representation that can capture important geometric and topological characteristics of an object/image. It has also been recognized as an alternative representation to facilitate DFM-related applications, such as reduced-dimension finite element modeling, feature extraction, and manufacturability evaluation (Hjalmarsson et al., 1994; Chu and Lee, 1993; Unruh, 1992; Patrikalakis and Gursoy, 1990; Prinz and Chern, 1988).

In general, there are two operations can be used to generate a skeleton. The first category is the medial axis transform (MAT) pioneered by Blum (1967) for biomedical applications. In this model, the transformation of an object to its skeleton is analogous to the propagation of a grass fire. If we consider a fire that is set all around an object's boundary and simultaneously progressing inwards at a constant speed, the skeleton is the place where the wave fronts meet (Blum, 1967). Equivalently, a 3-D MAT skeleton can be described as the locus of the centers of the maximum inscribed spheres. In addition to the medial axis/surface, a radius function is also associated with each skeletal point to represent its minimum distance to the boundary. Basically, the transformation to the MAT can be done by solving algebraic or differential equations to obtain the continuous skeleton (Unruh, 1992).

Most of the previous effort concentrated on the 2-D domain, and some progress has been reported in generating the continuous MATs for planar cross sections connected with lines and circular arcs (Patrikalakis and Gursoy, 1990). However, it is difficult for this type of transformation to cover all the different curves and surfaces in 3-D complex geometries. Attempts have also been made to translate the concept of MAT into the discrete world by utilizing discrete metrics and distance transformation (Montanari, 1968). However, besides being sensitive to boundary perturbation (Patrikalakis and Gursoy, 1990), the skeleton generated by this transformation usually contains too many "sub-branches" (Figure 3.12). In fact, there will be a "sub-branch" extended to every concave vertex, which significantly complicates the extraction of thin wall regions.

Furthermore, as implied in its definition, MAT is not necessarily a topology-preserving operation, and the results do not well match the geometrical features of the object. Therefore, it is difficult to use the MAT skeleton in this application unless a post-pruning operation is performed on the skeleton.

The second category, the thinning skeleton, conforms more to the desired medial surface for diecastability evaluation than does the MAT skeleton. Thinning is a topology-preserving data compression technique commonly used for digital images. Basically, thinning of an object is achieved by continuously deleting "non-critical" points from an object's boundary until the final skeleton is left. Thus, the resulting skeleton usually lies along the medial surface/axis of an object, and does not contain most of the undesirable "sub-branches" as opposed to the MAT skeleton.

Over the past decades, many thinning algorithms have been proposed, though most of them concentrated on 2-D (refer to Lam et al., 1992; Pavlidis, 1982 for comprehensive surveys). Recently, much attention has also been devoted to the generation of 3-D thinning skeleton (e.g. Ma, 1994; Saha and Chaudhuri, 1994; Lee et al., 1994; Kong and Rosenfeld, 1989). There is, however, no universally accepted 3-D thinning algorithm.

Figure 3.12 The Undesirable Sub-Branches in the MAT

3.2.3 The Generation of 3-D Thinning Skeleton

The major concerns in generating the thinning skeleton are the preservation of the topology and the retention of the end points. For topology preserving, the term "simple point" has commonly been used to denote those points whose deletion will not affect the topology or connectivity (e.g. Kong and Rosenfeld, 1989; Tsao and Fu, 1982; Srihari et al., 1979). On the other hand, the term "end point" is used to denote points that should not be deleted due to geometric consideration. This condition is used to preserve the geometric features of the object in the final surface/axis-like skeleton. In this section, we will discuss how these two conditions can be defined, and how they can be implemented efficiently.

3.2.3.1 Basic Definition

In this section, general definitions and notations of relevant digital topology are given to facilitate our discussion. As discussed before, binary voxels corresponding to a part are stored in a large 3-D Frame Buffer R, where each voxel is represented by a 3-tuple (i,j,k). A black voxel is the voxel that belongs to the part, while a non-occupied voxel is called a white voxel.

If ψ represents the occupancy state of a voxel, object S contains all the black voxels and can be expressed as: $S = \{V(i, j, k) \mid \psi(V(i, j, k)) = 1\}$, while $\bar{S} = R - S$ represents the set of all white voxels. Each voxel has three types of neighbors: 6-neighbors are voxels that share a common face, 18-neighbors are voxels that share at least a common edge, and 26-neighbors are voxels that share at least a common vertex. This is shown in Figure 3.13. In the following discussion,

$$N_{26}(V(i, j, k)) = \{(x, y, z) \mid \max(|x-i|, |y-j|, |z-k|) = 1\}$$

denotes all the 26-neighbors (vertex-adjacent) of a voxel V, while

$$N_{26}'(V(i, j, k)) = N_{26}(V(i, j, k)) \cup V(i, j, k)$$

is the set that consists of voxels in $N_{26}(V)$ plus the voxel V itself.

A black (white) voxel $V(i, j, k)$ is a 6-border voxel (also called a border voxel), if at least one of its 6-adjacent neighbors is white (black). In other words, if $V(i, j, k)$ is a black voxel then it is a border voxel iff:

$$\begin{aligned} & ((\psi(V(i+1, j, k)) = \phi) \vee (\psi(V(i-1, j, k)) = \phi) \vee (\psi(V(i, j+1, k)) = \phi) \\ & \vee (\psi(V(i, j-1, k)) = \phi) \vee (\psi(V(i, j, k-1)) = \phi) \vee (\psi(V(i, j, k+1)) = \phi)) \end{aligned}$$

These border voxels are further classified into E(East), W(West), N(North), S(South), F(Front), and B(Back) border voxels, respectively, depending on the directions (see Figure 3.14) in which its neighboring voxels are white (black). An interior black (white) voxel, on the other hand, is a voxel whose 6-adjacent neighbors are all black (white).

Figure 3.13 The 6-, 18-, and 26-Connected Neighborhood

Figure 3.14 Directions Used for the Classification of Border Voxels

To avoid a paradoxical situation, a different connectivity is used for S and $\bar{S}$ (Kong and Rosenfeld, 1989). In our application, the black component denotes the part, so 26-connectivity is used to ensure continuity. On the other hand, 6-connectivity is applied to white components to avoid ambiguity.

An n -path is a set of voxels $p_1, p_2, \dots, p_{i-1}, p_i, \dots$ where every consecutive pair of voxels are n -neighbors. Thus, two voxels are n -connected if there is at least one n -path between them. We will represent 26-connectivity between two voxels A and B by $A \Leftrightarrow B$. Based on this definition, an n -component is a set of voxels in $S(\bar{S})$ where all the voxels are n -connected. In our application, there is only one 26-connected black component, i.e. the part, and there is at least one 6-connected white component, i.e. the background. The background is that component of $\bar{S}$ that contains the boundary voxels of the volume, i.e. $\{V(i, j, k) | i = i_{\max} \text{ or } 0, j = j_{\max} \text{ or } 0, k = k_{\max} \text{ or } 0\}$. Any other white components that are completely enclosed by the object (in the sense of 6-connectivity) are called cavities. This is analogous to a hole in 2-D. A tunnel, on the other hand, can be defined as being topologically equivalent to a solid handle or a hollow torus. More detailed discussion of digital topology can be found in Kong and Rosenfeld (1989).

3.2.3.2 Simple Point

As stated before, a simple point is used to represent those points that can be deleted from an object without changing its topology. In this section, simple points will be discussed in detail. We will first discuss the concept of Euler characteristic, an important property in 3-D topology, and the main criterion for determining if a point is simple. Next, the various definitions of simple points, including ours will be discussed. Finally, we will present an efficient algorithm to identify the simple points.

The Euler characteristic was originally defined in continuous space to check the topological validity of solids. Basically, for a polyhedron, Euler's formula can be written as:

$$V - E + F = 2 - 2G \quad (3.3)$$

where V is the number of vertices, E is the number of edges, F is the number of faces, and G is the number of Genus. Genus can be defined as the largest number of non-intersecting simple closed curves that lie on a closed surface without separating it (Tsao and Fu, 1981; Meserve, 1955). It is in some sense similar to the notion of number of "handles". The number $V - E + F$ is called the Euler characteristic of a closed surface and is well known as a topologically invariant property (Meserve, 1955). If S is a set of polyhedra, the Euler characteristic, $\chi(S)$, can be described as:

$$\chi(S) = O(S) - H(S) + C(S) \quad (3.4)$$

where $O(S)$: number of objects,
 $H(S)$: number of holes (tunnels),
 $C(S)$: number of cavities (a white component surrounded by a black component).

To expand the Euler characteristic to the digital space, an analogous surface representation needs to be constructed. This involves representing the discrete voxel model by a set of surface patches. The basic principle is to retain the connectivity of the voxel model in its analogous surface representation. In this representation, a voxel is considered to be a block with 6 faces, and the faces, of the border voxels, that surround the object S and separate it from $\bar{S}$ form a surface patch of the polyhedra. In the case of a 6-connected object, a network of surface patches is constructed for the object, whereas in the case of 26-connected object, each 6-connected component of its complement $\bar{S}$ is represented as a polyhedron (Tsao & Fu, 1981; Lobregt, 1980). Based on this analogous surface representation, the Euler characteristic can then be computed for digital objects.

Since the Euler characteristic is a topologically invariant property, many thinning algorithms use it to detect simple points (Chu and Lee, 1993; Lobregt et al., 1980). In those approaches, if the deletion of a voxel changes the Euler characteristic, then it is not simple. It has been recognized that the effect of a voxel P on the Euler characteristic can be locally computed in a $3 \times 3 \times 3$ window, i.e. $N_{26}'(P)$. Lobregt et al. (1980) presented an efficient approach to compute the Euler characteristic by first dividing a $N_{26}'(P)$ into 8 overlapping $2 \times 2 \times 2$ octants. The Euler characteristic of the 256 voxel configurations of an octant is then pre-computed and stored in a look-up table. The Euler characteristic of $N_{26}'(P)$ is then computed by summing the contribution of each octant.

However, utilizing the Euler characteristic alone cannot guarantee preservation of the global topology. Figure 3.15 shows an example where the Euler characteristic is unchanged but the object can be separated by removing the center voxel P . This is because, in this case, the number of objects and holes increase by the same amount but the number of cavities in $N_{26}'(P)$ remain 0 because P is a border voxel.

There are other definitions of simple points used in the 3-D thinning. Srihari et al. (1979) used path connectivity of the black voxels to identify simple points. This is equivalent to saying that the number of black components remains unchanged after the deletion of the simple points. Even though path connectivity can be used to preserve topology in 2-D thinning algorithms, it is a necessary but not sufficient condition for 3-D voxel models. In 2-D (8,4) pictures, if the number of black components remains the same in a 3×3 window when the central border voxel is deleted, the number of white components is unchanged as well. In other words, this central border voxel can be classified as a simple point in 2-D. However, in 3-D, just checking the number of black components is not sufficient to guarantee invariance of the number of white components. Furthermore, a hole can be created even though the path connectivity for both black and white voxels is preserved. An example is shown in Figure 3.16 (Gong and Bertrand, 1990).

Figure 3.15 Example of the Euler Characteristic Unchanged but Object Separated by Removing the Center Voxel

Figure 3.16 Counter Example Showing the Problems of Using Path Connectivity to Preserve Connectivity (Gong and Bertrand, 1990).

A complete characterization of simple points has been given by Morgenthaler (1981).

- (a) $O(S \cap N_{26}'(P)) = O(S \cap N_{26}(P))$
- (b) $O((\bar{S} \cap N_{26}(P)) \cup P) = O(\bar{S} \cap N_{26}(P))$
- (c) $H(S \cap N_{26}'(P)) = H(S \cap N_{26}(P))$
- (d) $H((\bar{S} \cap N_{26}(P)) \cup P) = H(\bar{S} \cap N_{26}(P))$

In general, this statement means that a border voxel P can be classified as a simple point if and only if its deletion will not change the numbers of white components and white holes in a $3 \times 3 \times 3$ neighborhood, and the numbers of black components and black holes also remain the same.

Even though this definition of simple points seems plausible, the implementation of an efficient algorithm to count the numbers of objects and holes for both black and white voxels is very difficult. Therefore, there is a need for simplified criteria to identify the simple points. Since the Euler characteristic is topologically invariant and its computation is more efficient, it can be used to simplify the detection of simple points. With the introduction of Euler characteristic, these conditions were further simplified to:

- (a) P is adjacent to just one black component of $N_{26}(P) \cap (S - \{P\})$
- (b) P is adjacent to just one white component of $N_{26}(P) - S$
- (c) $\chi(S \cap N_{26}(P)) = \chi(\bar{S} \cap N_{26}(P))$

For the case of (26,6), conditions (a) and (c) imply (b) (Kong and Rosenfeld, 1989). The above simplification using the Euler characteristic was pioneered by Tsao and Fu (1982), but was also adopted by Kong and Rosenfeld (1989) and many others (e.g. Chu and Lee, 1993). Malandain and Bertrand (1992) proposed other simple criteria to select simple points. In their approach condition (b) is modified to:

- (b') P is 6-adjacent to only one white component in P 's 18-neighborhood.

In this way, simple points can be identified without using any connectivity number, so condition (c) can be dropped.

3.2.3.3 Implementation

In this research, we use the simple point criteria (a) and (c) because of the efficiency of computing the Euler characteristic. Besides the Euler characteristic, the only other requirement for the detection of simple points is to count the number of black components in $N_{26}(P)$. Most researchers tackle this problem by using modified flooding algorithms to connect voxels (Lee et al., 1994; Ranjan, 1991). Therefore, this is relatively time-consuming. Since the total number of voxels involved in the thinning process is significant, computational efficiency is a critical factor. We present a new approach to determine the number of black components.

The algorithm we have developed involves first dividing the $N_{26}(P)$ into 8 overlapping octants. For each octant, if any voxel in the octant is black, that octant is treated as occupied. Based on the possible occupancy states of the 8 octants, a lookup table containing 256 entries is built to expedite the process. For several octant configurations, the number of black components must be greater than one. In these cases, the voxels involved cannot be simple points, and no further testing is necessary. Figure 3.17 (a) lists these cases in terms of the sequence numbers of the octants (refer to Figure 3.17 (b) for the octant order).

Case 10011100, marked in figure 3.17(b), will be used to illustrate this. In this configuration, since octants 1, 2, 6, and 7 are off, voxel 3, 4, 6, 7, 12, 13, 14, 15, 16, 17, 21, 22, 23, 24, 25, and 26 must be empty. Therefore, the black voxel 8 must be separated from the rest of black voxels. For the cases that need further processing, the "flooding" process is performed on the 8 octants rather than on the 26 voxels. To facilitate the discussion, let us define the connectivity between octants. For this we shall use Figure 3.18 where each vertex corresponds to a voxel. Two octants joined by a common face, edge, or vertex are connected neighbors if and only if one of the voxels in the joining face, edge, or vertex is black. Two octants W_1 and W_n are connected if and only if there is a path $W_1, W_2, W_3, \dots, W_n$ where each consecutive pair of octants are connected neighbors. Based on this definition of octant connectivity, we can decide whether two adjoining octants can be tied together in the flooding process. The number of connected black components then equals the number of distinct groups of connected black octants. The efficacy of this algorithm is proved below.

Proposition 1: The black voxels within one octant form one 26-connected black component.

This is an obvious result of using 26-connectivity for the voxels.

Proposition 2: If two octants W_1 and W_n are connected neighbors, then the voxels involved in the two octants form one connected black component.

Proof : Let X be a common voxel between two octants where $y(X) = 1$

From (1) $\forall P_1 \in \{V \mid V \text{ is a voxel in octant } W_A \text{ and } y(V) = 1\}$ then by Prop. 1, $P_1 \Leftrightarrow X$

From (1) $\forall P_2 \in \{V \mid V \text{ is a voxel in octant } W_B \text{ and } y(V) = 1\}$ then by Prop. 1, $P_2 \Leftrightarrow X$

This implies $P_1 \Leftrightarrow P_2$

Corollary: If two neighboring octants are not connected, the voxels contained in the first octant are not connected to those in the second octant and vice versa.

Proposition 2 and its corollary are equivalent to saying that the establishment of the presence of a black voxel that is common to two octants is sufficient to show 26-connectivity between their black voxels. We can extend this to say that if W_1 and W_n are

connected, then all the voxels involved in the path form one connected component. The reason is:

If $P_1 \in W_1, P_2 \in W_2, P_3 \in W_3, \dots, P_n \in W_n$ where W_1 and W_2 are connected neighbors, W_2 and W_3 are connected neighbors,

$P_1 \Leftrightarrow P_2$ and $P_2 \Leftrightarrow P_3$ by proposition 2

$\therefore P_1 \Leftrightarrow P_3$.

By induction, we can extend this to W_n .

$\therefore \forall P_i, P_j \in \{W_1, W_2, \dots, W_n\} P_i \Leftrightarrow P_j$ and form one connected component.

By a similar argument, we can also say that if two octants are not connected in S , then the voxels contained within the two octants are not connected in S . This reasoning proves that our octant-based flooding algorithm produces the same results as counting the black components by connecting voxels.

Figure 3.17 Octant Configurations that Must Contain More Than One Component

Figure 3.18 The Voxel Configuration of the Example Case 10011100

3.2.3.4 End Point

The appropriate retention of the end points is a critical factor in determining the final skeleton. If this criterion is ignored, a thinning operation will shrink any sphere-like objects to a single point. Many different definitions of end points can be found in the literature. Depending on the intended application, some of the 3-D thinning algorithms result in axis-like skeletons (e.g. Chu and Lee, 1993; Tsao and Fu, 1982), while others generate medial surfaces (e.g. Gong and Bertrand, 1990; Tsao and Fu, 1981). It should be noted that if a voxel is retained, a skeleton branch will be extended to reach that voxel because of connectivity requirements. Due to the nature of binary voxel models, voxelized objects may appear jagged and blocky. Improperly defined end points may lead to the appearance of many noisy branches in the skeleton, which complicates the reasoning procedure. Therefore, one of the challenges is to develop a means to keep unnecessary noisy branches at a minimum while still retaining the important shape characteristics.

Our definition of end point is inspired by the thin octant concepts proposed by Morgenthaler (1981) and Lee et al. (1994). Morgenthaler first used the concept of "thin" octant to define the surface point condition. An octant Q of $N(P)$ is thin if P and every black voxel, inside Q and directly adjacent to P , is 6-adjacent to a white voxel of Q . P is an end point if all the octants of P or any of P 's 6-neighbors are thin. Thus, this approach requires checking of the $5 \times 5 \times 5$ neighborhood in order to identify an end point.

Lee et. al. (1994) have modified Morgenthaler's thin octant criteria to include only two regularized surface patterns and octants that contain at most three black voxels. A voxel

is called an end point if all of its eight octants are thin. Therefore, only the 3x3x3 neighbors need to be checked. In our approach, we seek a compromise between these two definitions. We include those patterns that have 4 voxels in an octant as thin octants except for the cases shown in Figure 3.19. It is because in all the other cases with 4 voxels, even though the surface does not form a regular pattern, such as the ones in Lee et al. (1994), the octants are still thin. An example is shown in Figure 3.19. In addition, we have added a few "wedge" patterns (see Figure 3.21 (a) and (b)) to the octant table. These are cases where the octant is actually thin even though as many as 6 voxels can be contained in one octant. Figure 3.21(c) shows the wedge patterns for the cases when there are 6 voxels in one octant, namely 11011101, 10111011, 10101111, 11110011, 11001111, and 11110101. The wedge patterns for the cases when there are 5 voxels in the octant can be obtained from the 6 voxel wedge patterns by changing one of the black voxels, other than P, to white. All the other patterns that do not fall into the categories mentioned above are thick. A voxel is not a surface point if any of its 8 octants is thick. Furthermore, if any of the non-thick octants belong to the 6 wedge classes, an additional condition, i.e. P is adjacent to two white components in $N_{26}'(P)$, was used to decide whether P is an end point. In conclusion, our characterization of end point can be stated as:

Let $\mathcal{R} = \{ W(P) \mid \langle W(P) \rangle \leq 4 \text{ and } W(P) \neq 11101000 \}$ where $\langle W(P) \rangle$ is the number of voxels in octant W(P).

The voxel P is an end point iff

1. $W_i(P) \in \mathcal{R}, \forall i \in \{1, 2, \dots, 8\}$, or
2. $\forall j \in \{1, 2, \dots, 8\} \ W_j(P) \in \{\text{wedge or } \mathcal{R}\}, \exists i \mid W_i(P) \in \{\text{wedge}\}$, and the number of white components adjacent to P in $N_{26}'(P)=2$

Figure 3.19 Voxel Pattern that Is Not Classified as a Thin Octant

Figure 3.20 Example Showing an Irregular Thin Pattern

(a)

(b)

(c)

Figure 3.21 Wedge Patterns that Are Thin Octants

3.2.3.5 Sequential vs. Parallel Approach

It is well known that the simple points criteria listed above are not sufficient to preserve connectivity in a true parallel algorithm. There are many situations that either the object will be separated or it will be annihilated. The most common approach to remedy this problem involves subdividing the parallel algorithm into 6 sub-iterations (e.g. Morgenthaler, 1981; Tsao and Fu, 1982). In each sub-cycle, border points of a specific direction (N, S, E, W, F, or B) are processed in parallel. However, even this directional parallel approach cannot completely solve the problem. A simple example is shown in Figure 3.22 where all the voxels are N-type border voxels, and they can be deleted altogether in one sub-cycle. Some researchers rely on the end point conditions to prevent this problem from occurring (e.g. Morgenthaler, 1981). But as stated before, end point conditions are used to retain the important geometric features and to prevent the skeleton from over-shrinking. In fact, end point conditions are expressly intended for this purpose and do not necessarily enforce connectivity. Further, the end point condition that can enforce connectivity does not necessarily produce the best skeleton. Therefore, the directional parallel approach does not always work.

Ma (1994) proposed a two-phase parallel thinning algorithm that checks all the deletable voxels in the second phase to see if the aforementioned problem occurs. If it occurs, some of the "deletable" voxels are kept according to a preserving rule. Lee et al. (1994) use a sequential re-checking approach achieve the same result. Since a parallel machine is generally not available, a sequential algorithm is used instead to detect simple points. This approach, however, suffers from significant noise for complex geometries because numerous points are unintentionally recognized as end points. The directional parallel algorithm produces a better result, but much noise still exists in the final image. In addition, the resulting skeleton is very sensitive to the order of processing directions.

For the directional parallel approach, there are several points worth emphasizing. First, many border voxels can be classified into more than one type of border voxels. If one of the directions is randomly assigned, many problems can occur. Figure 3.23 shows a simple example where voxel P sticks out of the boundary of a surface.

Figure 3.22 Annihilation of Voxels in Directional Parallel Thinning

Figure 3.23 . The Effect of Processing Orders on the End Point Detection When Using the Directional Parallel Approach

Even for a synthesized object, this type of boundary noise can be introduced during scan conversion due to rounding off errors. In general, the skeletonization algorithm would rather not extend a whole new branch to this point. In fact, that is part of the reason we test the number of white components for this wedge pattern in the end point criteria. However, undesirable results can happen if the point is processed using the wrong direction. In the example, point P can be classified as an N-, a W-, or an E- type border voxel. Suppose the order of directions chosen for processing are N, W, E, Then, if Voxel P is regarded as type W or E, it will be identified as an end point. To avoid this problem, multi-type border voxels should be processed in the earliest direction possible. In the example, P should be processed in the N-subiteration.

Figure 3.24 Example Showing the Differences of the End Points Detection by Using the Directional Parallel and the Complete Parallel Algorithms

Figure 3.24 shows another example of points being unintentionally identified as endpoints in the directional parallel approach, but not when border voxels are considered all together. In this example, if the directions of processing are as shown, the voxel P will be picked up as an end point. However, for most cases we would rather consider this geometry (a 2-voxel wide protrusion) to be a surface perturbation rather than extend a skeleton branch to that region.

In general, due to the unavoidably sequential nature of the directional parallel algorithm, it is very difficult to completely eliminate this problem. Our current approach involves checking the end point conditions for the surface in parallel (true parallel) first, and then deleting the simple points from the non-end-point set sequentially. Once a voxel is recognized as an endpoint, it will be retained and no further testing will be performed on it. However, this procedure does not give us desirable results when a two-voxel thick surface is left because none of those voxels will be recognized as surface points by the algorithm.

As an illustration, consider a flat plate. If there are an even number of voxels in the "thickness" direction, it is possible for only a single voxel to be left rather than the medial surface. To eliminate this problem, we first use a procedure to prevent over-shrinking. This procedure searches for a two voxel-thick pattern from all the border voxels in each of the three major directions. Once the pattern has been identified, the voxel with the larger distance value will be removed from the border list while the other is retained. Specifically, let us represent the south neighbor of voxel P as $S(P)$, and the north neighbor by $N(P)$, the east neighbor by $E(P)$, and so on. Then if P is a north border voxel and if $S(S(P))$ is white, i.e. $S(P)$ is a south border voxel, the border voxel (P or $S(P)$) with the larger distance value will be removed in the current iteration. The same

process is repeated for east and front border voxels as well. Only the border voxels that have not been removed will be processed in the current iteration.

3.2.4 An Alternative Distance-Based Approach

The thinning skeleton described above can be used to identify the thin sections of an object. However, the thinning process obviously takes more time than computation of the distance transform or any other aspect of the analysis process. Here, we describe an alternative approach that can approximately identify the thin sections. Even though this approach may occasionally fail to extract the medial surface and hence the thin section, it is very efficient and works very well for many practical cases.

The approach is completely based on the distance transform data of a 3, 4, 5 chamfer metric and only one traversal of the volume is required. The chamfer metric was chosen because it is orientation independent. If we look at the distance transform as analogous to height on a topographic map, the skeleton will correspond to the ridge lines, and can be extracted through identification of the local maximum. Local maximum means no other voxel in a voxel's 26 neighborhood has a larger distance value. As a consequence, this approach will fail whenever there is a height decrease in the ridge. However, unless the decrease is monotonic, it will be possible to identify the major shape of the ridge-line. In die casting, sharp corners are replaced with fillets and rounds and even when draft is added, the angles are relatively small. Based on our experience, this approach can handle a large percentage of the production parts.

3.3 Summary

In this chapter, the geometric characteristics that are often responsible for die casting thermal-related problems were first identified. These include thick and thin sections of a part and thin die regions. A procedure to extract the geometric characteristics from a voxelized object using inscribed spheres to model the wall thickness was described. Identification of these geometric characteristics involves the use of the distance transform and 3-D thinning. The distance transformation computes for each voxel of the part and the die its shortest distance to the part boundary. The boundary voxels are identified at the voxelization stage when the boundary is scan-converted. The distance transform value represents the radius of the inscribed sphere, centered at the voxel, that just touches the part boundary. The thinning skeleton is then used to label the voxels which lie near the medial surface of the part/die. Based on this model, the heavy sections can then be identified by extracting voxels with relatively large distance values because they are farther away from the part boundary. Thin regions of the part and of the die can also be extracted by identifying the medial surface voxels associated with smaller distance transform values. In addition, algorithms that can perform the analysis efficiently were described.

4. VISUALIZATION

4.1 Introduction

There has been much research effort in the area of scientific visualization to present multi-dimensional data sets (Crawfis, 1997). This has resulted in various techniques for presenting physical parameters that are the result of numerical simulations or experiments. The ideal tool should require a minimum of effort by the user and yet be able to present the analysis results with no ambiguity. It should help users quickly isolate regions that are potential problem regions and present the manufacturing information in a natural and understandable manner.

In the following sections, we will first discuss the volume rendering techniques used in this research, and the visualization methods to support the identification of the geometric incompatibilities in die casting.

4.2 Template-Based Ray Casting with Space-Leaping

For better perception and comprehension, the analysis results should be displayed directly in 3-D. In this research, we developed visualization tools that allow designers to view the problems regions, e.g. thickest sections, inside a semi-transparent object (like an X-ray film). This not only lets the designers view the regions of interest in perspective with the rest of the results but also helps them find the locations of these regions relative to the 3-D geometry.

Part of the goal is achieved through the use of volume ray casting. The basic algorithm of volume ray casting is shown in Figure 4.1. In this approach, a series of rays, which are either parallel to the viewing direction (parallel projection) or passing through the eye (perspective projection), are shot from each pixel on the screen window to the objects in the world space. For each ray, the algorithm calculates the point it enters the volume and the point it leaves the volume. It then computes the form of the ray based on these two points. As a ray passes through the volume, the voxel values are sampled along its length to obtain the color and intensity. This compositing process is continued until the ray accumulates enough opacity, hits an opaque voxel, or leaves the volume.

There are generally two different ways to generate the rays. The integer-based approach is to generate the rays by using a discrete line algorithm (Kaufman and Shimony, 1986), and to collect the opacity from the nearest voxel along each step of the ray. Alternatively, a floating point ray can be generated in a continuous fashion, and the voxel value is then sampled based on a higher-order interpolation (Kajiya and Von Herzen, 1984). The floating point ray is usually used when the value across the spatial extent of each voxel is varied in a pre-defined fashion (Yagel and Kaufman, 1992). Depending on the length increment between the sampling points, it allows the same voxel to be sampled more than once.

Generally, the continuous ray has the potential to provide a more realistic picture, but it requires more computation time. In our application, the voxel model represents a part

design. This volume is binary and is obtained through the voxelization of a synthesized object. The geometric reasoning procedures, e.g. distance transformation and thinning, applied to the model are concerned about the location of problem regions, not about precise shape. Thus, the DT value stored in each voxel is considered to be constant and uniform. Unlike scanned gray-scale data sets, taking multiple samples of the same voxel or sampling the voxel values based on an interpolation does not improve the image quality significantly. Besides, the integer rays are much more efficient in terms of rendering time. To support interactive decision making in preliminary design, speed of the diecastability evaluation is a crucial concern. Therefore, the integer ray casting is used to support a quick display of the evaluation results.

```

Set All Pixels = BACKGROUND ;
For Each Pixel (i,j)
    (X0,Y0,Z0) = Point_Enter_Volume (i,j) ;
    (X1,Y1,Z1) = Point_Exit_Volume (i,j) ;
    Init_Ray(X0,Y0,Z0,X1,Y1,Z1);
    (X,Y,Z) = (X0,Y0,Z0) ;
    Init (cur_prop) ;                      /* color & opacity */
    WHILE ( Inside_Volume(X,Y,Z) && ! Opaque(cur_prop))
        vox_value=Sample_Volume (X,Y,Z) ;
        vox_prop.color, vox_prop.opacity =Transfer (vox_value);
        cur_prop = Compositing (cur_prop, vox_prop, Normal(X,Y,Z)) ;
        (X,Y,Z) = Next_Voxel_Alone_Ray

    Draw_Pixel(i, j, cur_prop.color);

```

Figure 4.1 The Basic Algorithm of Volume Ray Casting

Ray casting can support volume semi-transparency; however, it is relatively slow in its basic form. Even with the integer rays, it is still very time consuming to generate the rays for all the pixels on the screen window and to sample the voxel values along the rays. Several methods have been suggested in the past to reduce the computation time for volume-based ray casting (e.g. Yagel and Kaufman, 1992; Zuiderveld et al. 1992; Levoy, 1990). The template-based algorithm proposed by Yagel and Kaufman (1992) is one of them, and is adopted here as a basis to support the visualization activities in this research. In this method, the coherence among integer-based rays is utilized to expedite the parallel viewing process.

In volume ray casting, the rays are emitted from the centers of the pixels on the screen into the volume, but the entry points on the volume are usually not centered on the

voxels. Thus, even though the slopes of the rays in parallel viewing are the same, the actual forms of the rays might be different due to rounding errors (see Figure 4.2). The basic idea of the template-based approach is to make the form of the rays the same by casting the rays from each pixel on a base plane instead of the screen window (see Figure 4.2). The base plane is one of the six volume faces that has the largest projection area on the screen, and is closer to the viewer. In this way, instead of re-performing the line algorithm for each pixel, the form of the rays is calculated once and stored in a template. For the template, we have chosen the 26-ray, in which the two adjacent voxels are at least joined by a vertex, rather than the 6-ray, in which the two adjacent voxels must share a face. The 26-ray contains fewer voxels than the 6-ray, and more importantly it can sample the voxel space uniformly (Yagel and Kaufman, 1992). A series of parallel rays are then emitted from the base plane to the volume by following the template. After this "base plane ray casting," the projected image on the base plane is then mapped to the screen window.

Figure 4.2 The Basic Principle of the Template-Based Ray Casting

Space leaping of the rays based on DT is another approach to accelerate the volume-based ray casting. It was first proposed by Zuiderveld et al. (1992). Basically, in this approach the acceleration of ray casting is achieved by jumping through the volume according to the DT value at the current voxel. As discussed before, the DT value at each voxel indicates its closest distance to the designated voxels. If the distance

transformation is applied to the empty voxels relative to the object, each non-occupied voxel will contain a value indicating the distance (i.e. the number of steps) to its closest object voxel. If a voxel has a DT value N , it can be guaranteed that the ray can make a jump of N steps from this voxel without missing any object voxels. Thus, many unnecessary steps can be skipped. Since the distance transformation is used in the geometric reasoning process, the resulting DT data is readily available to support the expediting the rendering process.

The two methods described above were developed separately. However, when combining those two methods together, the rendering process can be significantly expedited. As stated before, the template stores the form of the ray for parallel ray casting. The pattern of rays can be stored in either of the two forms, i.e. incremental or cumulative, as shown in Figure 4.4. The incremental template stores, at each step, the increments (0 or 1) in x -, y -, and z -directions that should be used to reach the current voxel from the previous one along the ray. The cumulative template, on the other hand, is built by summing all the previous increments in the x -, y -, and z -directions at each step. In other words this template contains the total number of increments that must be taken in the x -, y -, and z -directions to reach the current voxel from the starting voxel.

For space-leaping, the cumulative template is used. By implementing space-leaping in conjunction with a template, the rendering process can be significantly expedited (Yagel and Ciula, 1993). Figure 4.4 shows a 2-D example to illustrate the space-leaping process. Note that the distance transformation in this example is based on the chess-board metric.

One of the criteria for space-leaping to work properly is the consistency between the metric used in distance transformation and the type of rays used in rendering. However, as discussed before, 26-connected rays are used in the rendering due to efficiency and uniform-sampling. On the other hand, the distance transformation for wall thickness evaluation is based on the city block metric (6-connected distance) or 3-4-5 chamfer metric. One step of the 26-connected ray may correspond to a city-block distance of 1, 2, or even 4. This incompatibility in metrics can cause over-jumping, and the rays can miss the object.

The strategy used in this research involved building a conversion table between the DT values (city block or 3-4-5 chamfer) and the 26-connected ray template. The conversion table for the city-block metric is constructed by repeating the voxel entries in the incremental template N times, where N is the sum of its x -, y -, and z -increments. In other words, at a particular entry of the template, if the next voxel is reached by incrementing x - and y - by one step, then this entry is repeated twice in the conversion table. For the case of the 3-4-5 chamfer distance, the same principle is used except each entry is always repeated two additional times. These are shown in Figure 4.4.

When the volume is ray cast with the space-leaping method, the DT values at the current voxel corresponds to the number of steps that can be skipped in the conversion table. To reach the position of the next entry of the template, a jump is made in the conversion table. This entry in the conversion table then points back to the template and the value there indicates the position of the next voxel to be sampled. Note that in each template entry, a pointer points to the first occurrence of that entry in the conversion table. For

example, the step number 4 first appears at position 5 in the conversion table (see Figure 4.5), so its index into the conversion table is 4. The pseudo code for this "template-leaping" algorithm is shown in Figure 4.6.

Figure 4.3 Template to Support Space-Leaping of the Ray-Casting

Ray Casting with Space-Leaping

Ray Casting without Space-Leaping

■ : object voxels

Figure 4.4 Space-Leaping Based on the Distance Transform

Figure 4.5 The Conversion Table to Accommodate the Difference in Metrics for Distance Transformation and Ray Casting

```

NextVoxelAlongRay(vox_pt, template_pt)
{
    if(!LEAPING)
        voxel_pt = entry_pt + template[template_pt++].xyz;
    else{
        jumps = DT[voxel_pt];
        conversion_pt = template[template_pt].conversion + jumps;
        template_pt = conversion_table[conversion_pt];
        voxel_pt = entry_pt + template[template_pt].xyz;
    }
    return;
}

```

Figure 4.6 The Template-Leaping Algorithm

4.3 Three-Level Scheme

The aforementioned ray casting algorithm significantly improves the rendering speed. In addition, it supports semi-transparency so that the problem regions can be displayed along with the object. However, a typical semi-transparent volume rendering is amorphous and blocky and the shape of the object usually cannot be clearly comprehended. This problem is especially serious for binary-models. For the geometric complexity in typical diecastings, this blurry display hinders the designer from precisely locating the problem regions. In this research, we use a three-layer-scheme to facilitate better perception of the geometric object and its problem regions.

In this model, the voxels in the part are divided into three different layers. The skin layer contains voxels on the boundary of the part and is usually one or two voxels thick. These voxels are assigned a gray color, and a see-through opacity. The second layer, i.e. the highlighted core layer, consists of the voxels with the characteristics of interest. In the case of the heavy sections, it contains the voxels of the part with a distance greater than the specified value. These voxels are totally opaque, and are highlighted with a red color. For the thin part sections, it contains the voxels on the part skeleton with DT values less than the specified distance. These voxels are also opaque but are assigned a green color. All the other voxels of the part constitute the intermediate layer, and are totally transparent.

There are two major advantages to dividing the voxels into three layers. First, it improves the visual presentation of the evaluation results. For the voxels of the skin

layer, their normal vectors can be inherited from the polygon model during the scan conversion process. By explicitly rendering those outer surface voxels with the nearly-correct normal vectors, the shape of the object can be better perceived.

In addition, different surface properties for the skin and the problem regions, especially the thick sections, can be experimented with to provide a depth-cue so the two can be easily distinguished. For example, the boundary of an object can be more shiny and visible by employing higher specular and diffusive reflection coefficients on the skin voxels. On the other hand, the "thick voxels" can be set to be less diffusive to give the appearance of being inside the object. This scheme is analogous to viewing a part design whose boundary is made of glass, and whose problem regions are embedded within the object. Two examples are shown in Figure 4.7 to demonstrate the results of this scheme. As can be seen from these semi-transparent views, the user can view not only the hidden surfaces but also the inside problem regions, which make the advantages of this scheme evident.

The second reason to use the three-layer-scheme is to improve the rendering efficiency. Since the voxels in the intermediate layer are completely transparent, they can be skipped by the ray so that shading and compositing at this layer is not necessary. Thus, the speed of the rendering can be increased.

In addition, with the partition of object voxels into three layers, the space-leaping can be applied inside the object voxels as well. This is because the DT value of each object voxel is its closest distance to the object boundary. Therefore, the ray can safely make a jump through the intermediate layer according to the DT value of the current voxel without stepping out of its boundary.

The only thing that the ray can possibly miss is the highlighted core voxels. Therefore, the original space-leaping algorithm must be adjusted to avoid this situation. If $V_{i,j,k}$ denotes the DT value at position (i,j,k) and H denotes the threshold value selected for the problem regions, the maximum number of steps that a ray can jump becomes $\text{Min}(V_{i,j,k} - 1, H - V_{i,j,k})$.

It should be noted, however, for the visualization of thin die regions, some of the voxels of interest do not belong to the object. These include voxels on the die skeleton with a distance value less than the threshold. They too are part of the problem regions, and will be opaque and highlighted in yellow. Space leaping of the non-object voxels is not allowed in this case, because the ray might over-shoot some voxels of interest, i.e. thin die region.

(a) Opaque View

(b) Semi-Transparent View of (a)

(c) Opaque View

(d) Semi-Transparent View of (c)

Figure 4.7 Example Demonstrating the Semi-Transparent Effect Produced by the Three-Layer-Scheme

4.4 Shading Model

As discussed above, the normal vectors on the skin layer will be inherited from the faceted model during the voxelization process. However, there is still a need for a normal estimation method of the voxels in the core, especially the thick sections, to convey the information correctly. Due to the discrete nature of voxel-based models and the interior problem regions, an appropriate shading model is especially important in this research to allow the user to perceive the heavy mass regions relative to the 3-D object. Generally, in a shading model, the color intensity at each pixel is computed based on the surface orientation at the point of intersection, the surface attributes, the light sources, and the eye position. In conventional surface-based models, normal vectors, i.e. the surface orientation, can be easily extracted from the part representation. In voxel-based models, however, the exact normal vectors are not well-defined.

In this section, we will discuss the various volume shading models, and then propose a new 3-D contextual shading algorithm that can support a better perception of the heavy regions in the 3-D perspective.

4.4.1 Previous Shading Models

Depth gradient is a shading method that is commonly used for binary models (Gordon and Reynolds, 1985) as one way to provide the depth-cues. The simplest model is called "depth only". It is based on the fact that humans perceive the light intensities relative to the distance of the object from the viewer. In other words, a voxel closer to the screen appears brighter than the voxels farther away. In this model, the normal vector at any visible voxel is computed based on the gradients of the depth values in its four neighboring pixels on the screen:

$$N_{i,j} = (N_x, N_y, N_z)$$

$$N_x = D_{i-1,j} - D_{i+1,j}; N_y = D_{i,j-1} - D_{i,j+1}; N_z = 1 \quad (4.1)$$

where $D_{i,j}$ is the depth value at pixel (i,j)

It is an image-space shading model and thus is usually coupled with other image space viewing methods, e.g. Z-Buffer. For an object space rendering algorithm, such as ray casting, it is slightly cumbersome to use. This is because in ray casting, the rays are shot by following the order of scan lines; therefore, the depth information of the neighboring pixels required for the determination of a voxel normal is not all immediately available for each ray. The appropriate color of a pixel can only be determined after the ray casting process, or at least after the depth values of its neighboring pixels are available.

In our application, template-based ray casting is used to expedite the rendering process (Yagel and Kaufman 1992). Since the rays are shot from the base plane rather than the screen, another transformation is required to obtain the depth information. In addition, the results of using this model are sometimes of poor quality due to the lack of the third dimension. In this case, the processing overhead makes it difficult to justify its use in our application.

There are some other shading models, e.g. contextual shading and gray-level gradients shading, which perform the normal vector estimations in the object space. The basic idea behind contextual shading is to determine the surface inclination of a voxel based on the pattern of surface voxels in the neighborhood. Chen et al. (1985) proposed a normal-based contextual shading model based on a cuberille volume representation. The normal vector at any voxel is estimated based on the voxel face and its four adjacent faces. Since only a fixed number of voxel face patterns can appear in a small neighborhood, the normal vectors can be pre-calculated and stored in a look-up table. This model results in only 25 distinct normals (Kaufman, 1990) which, in general, is not sufficient to adequately display complex geometries. Within the context of the contextual approach, there are more sophisticated methods which fit a higher order surface through the voxel neighborhood (Yagel et al., 1994; Webber, 1990). However, the extensive computation will not be affordable in our application.

Hohne (1986) proposed another object space shading model that uses the gray-scale gradients as a measure of the normal vectors. It was originally used in gray-level images sampled with CT or MRI in medical applications. This basic idea of this algorithm is based on the fact that the normal vector of a surface ($F=c$) can be obtained by the gradient vector ($\partial F/\partial x, \partial F/\partial y, \partial F/\partial z$). Thus, we can use the discrete central difference in a voxel neighborhood to approximate the partial gradient. In other words, if we denote the voxel value at coordinates (i,j,k) as $V_{i,j,k}$, then

$$\partial F/\partial x = V_{i+1,j,k} - V_{i-1,j,k} \quad (4.2)$$

Similarly, $\partial F/\partial y$ and $\partial F/\partial z$ can be expressed in the same way.

Gray-scale shading produces a large, dynamic range of normal vectors, and can also support voxel semi-transparency. However, the basic assumption behind this algorithm is that the object being rendered exhibits a "partial volume effect". This means that the values of the near-surface voxels correspond to the relative averages of values of the different material types inside those voxels. Hence, in general, this algorithm is not applicable to binary voxel models. In our application, where the distance transform is available for the voxels of the volume, this approach is still not fully applicable. Figure 4.8 shows several 2-D cross sections and part of their distance transform. If the threshold value for the heavy sections set to 5, it can be seen that the gray-scale method gives us two different normals for cases (A) and (B), even though they should have the same normals. In addition, this method cannot distinguish case (B) from case (C), even though the surface inclination for both cases is different. Furthermore, when city-block metric is used for the distance transformation, the possible normal vectors resulting from this algorithm are about the same as the cuberille-based contextual shading model.

Figure 4.8 The Problems with Gray-Scale Shading of Voxel Model with Distance Transform

4.4.2 New Contextual Shading

As stated above, existing shading methods are not suited to presentation of manufacturing defects internal to the part such as heavy mass regions. The depth-based shading methods are too cumbersome to use with template-based leaping algorithms and are not accurate enough. The normal-based contextual shading proposed by Chen et al. (1985) can support only a very limited domain of normal vectors. On the other hand, Hohne's gray-scale shading (1986) is very sensitive to the distribution of the distance values in the sub-surface layer. Accordingly, we developed a new 3-D contextual shading method that overcomes these problems, and can be applied to the shading of binary voxel models.

An ideal contextual shading method for binary voxel models would involve hard coding a lookup table containing the normal vectors associated with all possible voxel configurations in a small neighborhood. In 3-D, if we restrict ourselves to $3 \times 3 \times 3$, this table will contain 2^{26} entries. Obviously, not only does this table have huge memory requirements, but also the computation and the input of the normal vectors for all the 2^{26} possible configurations are infeasible.

In our approach, we use a divide and conquer strategy to handle the unmanageably large number of configurations. It involves representing the $3 \times 3 \times 3$ neighborhood with three orthogonal planes, each with nine voxels as shown in Figure 4.9 (b), (c) and (d). Since each plane will then have $3 \times 3 = 9$ entries, the 2-D normal vectors of these 256 possible patterns can be computed and coded into a look-up table. At run time, the normal vectors from the three orthogonal planes are then combined to form the final normal (Figure 4.9 (e)). When building a normal table in the context of 2-D, there are many cases where the normal vector is not obvious. In those cases, multiple normal vectors are encoded and the appropriate one chosen based on the quadrant of the viewing direction. Furthermore, to account for the influence of the 8 vertex-adjacent voxels in the corners of $3 \times 3 \times 3$ neighborhood that have been neglected in the use of the 3 orthogonal planes, vectors diagonal to the $3 \times 3 \times 3$ cube and originating from each occupied corner voxel are added

together. The resultant can be normalized and combined with the vectors of the three orthogonal planes to provide a more accurate result.

Figure 4.9 Description of the 3x3x3 Contextual Shading Method

In this research, we also tried to improve the results by increasing the size of the neighborhood to $5 \times 5 \times 4$. Like the $3 \times 3 \times 3$ case, we divide the $5 \times 5 \times 5$ neighborhood into 3 orthogonal planes. However, since each plane contains $5 \times 5 = 25$ entries, creating a look-up table for all 2^{24} patterns is still impractical. Our strategy, therefore, is to reduce the number of patterns by systematic elimination of the entries that have little effect on the normal. In particular, since the normal is affected solely by the surface voxels, we can neglect all interior points.

The first task involves finding the voxels that have the most significant effect on the surface inclination. To accomplish this, the open direction(s) of a voxel are found based on whether that voxel (the center voxel in the $5 \times 5 \times 5$ neighborhood) is a North, South, East, West, Front or Back border point, i.e. its North, South, East or West neighbors are unoccupied. A voxel is called X-open if it belongs to that class of X-border points but not to its converse. For example, if the central voxel is a South border point but not a North border point, then the South is an open direction as shown in Figure 4.10. If, on the other hand, the central voxel is both a South border point and a North border point, then neither North nor South is an open direction. If all of its 4 direct neighbors in any plane are occupied, then its 4 indirect neighbors in that plane (voxels joined by only a common edge) will be checked. However, if all 4 indirect neighbors are also occupied, the plane has no contribution to the 3-D normal at that voxel.

Figure 4.10 Example to Illustrate the $5 \times 5 \times 5$ Contextual Shading

For each open direction, the 5x5 plane is then divided into five 5x1 columns that are parallel to that open direction. Each column is then traversed from the open direction to find the first surface voxel encountered. If a surface voxel is encountered in a column, it has at most 5 possible locations, or the surface does not pass through the column. Since only the first surface voxel of each of the 5 columns is used, the number of possible patterns is reduced to $6^5=7776$.

Even though this is still a relatively large number, many cases can be neglected. For example, since the center voxel of the 5x5 neighborhood lies on the surface, the first surface voxel in the center column cannot possibly be located beyond the center voxel. This effectively reduces by half the total number of possible patterns. In addition, if there are no surface voxels in the second or fourth column, the surface must be practically parallel to the length of the columns so any surface voxels in the first or fifth column, respectively, have no influence on the normal and can be neglected. Therefore, the effective number of combinations for the 5x5 plane is reduced to less than 1000. Based on this reduced set, a normal lookup table can be constructed with little effort.

Figures 4.12-4.14 show images rendered using the newly developed contextual shading methods. These results are compared with polygon model (Figure 4.11) and the gray-scale gradient shading applied to the gray-level versions of the object (Figure 4.15, 4.16). The fuzzy objects are generated by applying a 3x3x3 and 5x5x5 Gaussian filtering (Russ, 1995) to the original binary object so it is more time consuming. Figures 4.12(b) -4.16(b) also show the difference between these images and the image generated with the polygon normals. The darker shades of gray, in these figures, indicate larger error. As can be seen from the figures, though all of the methods produce promising results, the 3x3x3 contextual shading with the corner effects is slightly better than the others, including the gray-level object. It is probably because the 3x3x3 contextual takes care of all the ambiguous cases so the results are more accurate near the corners of the object. Since these new contextual methods can display an object practically in real time, without the need to store a duplicated fuzzy object, they have a significant advantage over the filter-based approaches.

Figure 4.11 An Object Shaded with Inherited Polygon Normals

mean: 0.0337

Standard Deviation: 0.0518

(a) The Shaded Image

(b) The Difference with Figure 4.12

Figure 4.12 The 3x3x3 Contextual Shading Method

mean: 0.0295

Standard Deviation: 0.0460

(a) The Shaded Image

(b) The Difference with Figure 4.12

Figure 4.13 The 3x3x3 Contextual Shading Method with Corner Effects

mean: 0.0376

Standard Deviation: 0.0573

(a) The Shaded Image

(b) The Difference with Figure 4.12

Figure 4.14 The 5x5x5 Contextual Shading Method

(a) The Shaded Image

mean: 0.0396

Standard Deviation: 0.0615

(b) The Difference with Figure 4.12

Figure 4.15 Gray Scale Gradient Method Applied with a 3x3x3 Gaussian Filter

(a) The Shaded Image

mean: 0.0304

Standard Deviation: 0.0525

(b) The Difference with Figure 4.12

Figure 4.16 Gray Scale Gradient Method Applied with a 5x5x5 Gaussian Filter

4.5 Presentation of the Analysis Results

The extraction and display of the problem regions described above depend heavily on the threshold value chosen for the highlighted core layer. Let us use the heavy section as an example. If the value used is too small, a red hot mass will be shown everywhere in the part. On the other hand, if this value is too large, the core will simply miss potential heavy masses. Using the average distance value of all of the voxels in the part as an index provides one way to deal with this problem. It is more flexible than the traditional rule-of-thumb evaluation because the value is dynamically adjusted based on the global characteristics of the part. Since the values used for averaging are the distances from the voxels to the boundary, this method is probably better than the average wall thickness estimation used in typical B-Rep models, i.e. volume divided by surface area. Another slightly better method to attack this problem is to derive the threshold based on the average DT of the skeletal points only rather than of the whole part. However, thick and thin are actually relative terms. Their presence depends on the global shape configuration as well as on its local neighborhood. Obviously, it is difficult for these two methods to account for local shape variations.

In this research, animation is utilized in our visualization module to provide a better way to facilitate the identification of the problem regions. To evaluate a die casting, the designer can specify the range of threshold values that are of interest and the rotation angle between each frame in the animation. This not only provides another depth cue resulting in better perception of the geometry, the continuously varying distance thresholds also help designers better perceive the problems.

For the animations, the highlighted red regions will appear to shrink or erode while the part surfaces remain semi-transparent to provide geometric perspective. The eroding red regions are the relatively thick sections in terms of the specified value, and their relative bulkiness indicates the severity of the heavy section problem. Furthermore, the breaks in the red core pattern indicate non-uniformity in wall thickness. The places where the red regions vanish and turn completely gray are the thin regions. If the 3-D thinning analysis is performed, once the shrinking core reaches the medial surface, the skeletal voxels, i.e. thin sections, stand out explicitly with a green color. This can help the user better comprehend the occurrence of those breaks, i.e. thin section problem. In this way, the regions that are the first to turn green are the thinnest regions while the regions where the core persists the longest are the thickest.

Figure 4.17(a) shows the analysis results of a part that was developed at OSU for die casting studies. This part has been modified by making a portion of the wall thinner to facilitate the illustration. From the sequence of images (b-c), the thinnest (region that first turns green) and the thickest regions (the red regions that persist the longest) can be easily identified.

The user also has the option to perform thinning analysis for the die, and give different threshold values for thin die regions, in which case the thin die regions will be displayed in yellow. In addition, the boundary of the object can be turned on or off. In this way, the

user can see the thickness results without being obstructed by the object boundary, while still having the ability to precisely locate the problems with respect to the part.

The scheme described above facilitates the visualization of the problem regions in the perspective of the overall part characteristics. In other words, the characteristics of the problem regions can be compared with those for the rest of the part to determine the severity of the problems. The attributes that are most important are the sizes of the highlighted regions, and the interactions among the various highlighted regions. Specifically, the user can find interactions like those between thick sections and thin die regions that exacerbate thermal-related problems.

4.6 Results

Based on the approach described above, a prototype system was implemented. This system accepts stereolithography (SLA) files as input and generates the voxel model from it. Upon the user's command, the system will evaluate the part design using distance transform and 3-D thinning skeleton. If required, it will also generate a series of motion pictures to help users visualize the problem regions. The reasoning mechanism has been designed to be transparent to the user and simple to use. This system is CAD platform independent and can be integrated with any open architecture CAD system that supports polygon models.

Figure 4.18 - 4.20 show several examples of die casting parts and the analysis results. In Figure 4.18, besides showing the thinnest section in (a), the sequence of images (b-d) shows the relative thicknesses for the part. The sequence is analogous to peeling off layers of an onion, resulting in the display of the heavy sections that can be naturally visualized by designers. Figure 4.19 and Figure 4.20 are two more examples.

The approximate time required in each phase of the analysis is listed in Table 4.1. These results are obtained by running this system on an IBM R6000 37T workstation with 128 MB memory and Power Gt4e Graphics Adapter. As can be seen from this figure, the distance transformation with either the 3-4-5 chamfer metric (quantitative analysis) or the city-block metric (qualitative analysis) is extremely efficient, and the time required is approximately proportional to the volume size, i.e. number of voxels. In addition, the qualitative analysis is about 2-3 times faster than the quantitative analysis. This is because all the 26 neighboring voxels of any given voxel are required for the computation if 3-4-5 chamfer metric is used while only 6 direct voxels are used for the computation of city-block distance.

For the distance-based thin section analysis, the time required is also proportional to the size of the volume. Again, this is because in this approach, the volume is scanned once to identify the voxels with a local maximum distance value. For voxelization, the time taken depends on several factors, including the number and size of polygons, volume size, and the shape complexity. As can be seen from table 4.1, the case shown in Figure 4.18 takes the shortest voxelization time because the shape is relatively simple and it has the smallest number of polygons and smallest volume. On the other hand, the case shown in Figure 4.20 takes the longest voxelization time because the shape is more complex and it

has a relatively large volume and a large number of polygons. A more detailed analysis can be found in Ouyang (1994).

(a) CAD Model

(b) Thin Section (c) Thick Section

Figure 4.17 An Example Showing the Identification of Thin and Thick Sections

For thinning, besides the number of voxels, the time required depends on the number of iterations required to reach the final skeleton, and the number of voxels processed in each iteration. Thus, a thin shell object, e.g. a hollow ball, is relatively fast to generate the part skeleton. However, it will take significant time to generate the die skeleton. When combining the two together; i.e. the thinning for die and the thinning for part, the total number of voxels will have the most significant effect on the total time required as shown in Table 4.1.

Part	Volume	# of voxels	# of polys	Qualitative DT	Quantitative DT	Thinning -Part	Thinning - Die	Distance-based
Fig 4.17	150x76x143	1630200	344	12 sec	34 sec	114 sec	-	10 sec
Fig 4.18	150x47x66	465300	242	4 sec	10 sec	19 sec	97 sec	4 sec
Fig 4.19	104x150x10 4	1622400	760	13 sec	34 sec	87 sec	285 sec	10 sec
Fig 4.20	150x52x117	912600	898	7 sec	18 sec	135 sec	120 sec	6 sec

Table 4.1 Computation Time for the Various Phases of the Analysis

In general, the visualization system provides results relatively quickly, and the presence and location of the problem regions can be easily perceived. Compared with typical numerical simulation which takes at least an hour, this system is significantly more efficient.

(a) The Thinnest Section

(b) The Heavy Section (Sequence 1)

(c) The Heavy Section (Sequence 2)

(d) The Heavy Section (Sequence 3)

Figure 4.18 Results of Wall Thickness Evaluation (Part 1)

(a): 3-D Sectioned View

(b): Thin Die Region

(c) : Thinnest and Heaviest Sections

Figure 4.19 Results of Wall Thickness Evaluation (Part 2)

(a) Opaque View

(b) The Heaviest Sections and the Thin Die Region

Figure 4.20 Results of Wall Thickness Evaluation (Part 3)

5. VERIFICATION OF THICK/THIN ANALYSIS

5.1 Introduction

Verification of the system and testing of its performance involves comparing the analysis results for selected designs with actual production problems and the results obtained from numerical simulations. The research is based on the abstraction of design guidelines and empirical experience. Specifically, the abrupt changes in wall thickness of the part and the geometrically confined thin regions of the die can cause defects and should be eliminated wherever possible. The examples in the previous chapter demonstrated that this system correctly identifies these geometric characteristics. The objective of this chapter is to verify that this geometry-based approach corresponds to data provided by numerical simulation and does not present misleading results.

For the case of thin die regions, most diecasters know that they will cause die maintenance problems no matter what corrective action is taken. For the case of wall thickness variation of the part, especially the thick section, we will show the value of the system by comparing the analysis results with solidification simulation results of a commercial FDM package, MAGMAsoft. Furthermore, we will relate the heaviest section with the last region to solidify. Finally, we will use a redesign example to demonstrate the usefulness of this system. It should be noted that some of the geometric incompatibilities are often accommodated by employing expensive die-tryouts and sophisticated processing techniques in the production. For those cases, our claim is that the use of this system can significantly reduce the design iterations originally required.

5.2 Case Studies

In the following case studies, we will first show the analysis results from the visualization system and compare them with the temperature profile of a solidification simulation obtained by using a numerical tool, i.e. MAGMAsoft. For a more quantitative comparison, we have also sampled several points in the heaviest section of the object and placed "thermocouples" at those points in the numerical simulation. For the cooling curves obtained for these points, an intersection line can be drawn corresponding to the time at which approximately 95% of the metal has solidified (obtained from the message window near the bottom of the MAGMAsoft results). We can, therefore, find the states of the heaviest regions at the time when 95% of the volume has solidified.

5.2.1 Case 1

The part in Figure 5.1 is one produced by a North American Die Casting Association (NADCA) member. The NADCA technical staff was asked for advice on how to address the porosity problems which were readily apparent when the part was sectioned (see Figure 5.1). The same part was tested against our diecastability evaluation program. The heavy sections were found near where the boss joins with the rest of the part (see Figure 5.2), which coincide with the regions of porosity in the actual part. To remedy this porosity problem, the most straightforward solution is to core out the heavy boss, so that the wall thickness can be more uniform.

Figure 5.3 shows the temperature profile of numerical simulation from MAGMAsoft. It can be seen from the two figures that the thinnest section and the thickest section respectively correspond to the coolest regions and the hottest regions. Figure 5.4 shows the location of the points being sampled from the heaviest section, and the corresponding cooling curves are shown in Figure 5.5. As indicated in Figure 5.5, when 95% of the total casting has solidified, the metal at those points is just starting to release the latent heat. Therefore, the heaviest sections are located in the regions where the last 5% of the metal solidifies.

Figure 5.1 Sectioned Part Obtained from NADCA Member

(a) The Heavy Section and the Thin Section

(b) Another View of the Heavy Section

(c) Distance Distribution of the Object Near the Center of the Boss (A Sectioned View)

Figure 5.2 Analysis Results of the Visualization System (Case 1)

Figure 5.3 Results from Numerical Simulation (Case 1)

Figure 5.4 Location of Points Sampled from the Heaviest Section (Case 1)

Figure 5.5 Cooling Curves for the Sampled Points (Case 1)

5.2.2 Case 2

Figure 5.6 shows heavy sections for another diecasting that was obtained from industry. Because of porosity problems near the center of the part, many parts failed in the field. The company had to use a special mechanism to increase the density of the metal in the problem region for pressure tightness. Besides, additional capital had to be invested due to the special mechanism, and there was a high scrap rate. These problems were again caused by the design's not being compatible with the die casting process because it had a region that was too heavy. In fact, without changing the geometry, other manufacturing processes, such as squeeze casting, may be more suitable to produce this "thick" part. The visualization system helped us easily find this incompatibility, as shown in Figure 5.5. A comparison with the temperature profile of MAGMASoft simulation as shown in Figure 5.7 further illustrates that the thickest regions identified from the visualization system correspond well with the hottest regions.

Again, Figure 5.8 shows several points sampled from the heaviest region. As can be seen from the numerical simulation (Figure 5.7), when 95% of the casting has solidified, the time is approximately 3.44 seconds. From the cooling curves (Figure 5.9), we can see that the metal at those points is just starting to release the latent heat at that time. Therefore, the heaviest sections are located in the regions where the last 5% of the metal solidifies.

(a) The Relatively Heavier Region

(b) The Heaviest Region

(c) Distance Distribution of the Object Near the Heavy Section (A Section View)

Figure 5.6 Analysis Results of the Visualization System (Case 2)

Figure 5.7 Numerical Simulation Results (Case 2)

Figure 5.8 Location of Points Sampled from the Heaviest Section (Case 2)

Figure 5.9 Cooling Curves for the Sampled Points (Case 2)

5.2.3 Case 3

The part in Figure 5.10 was developed in-house for a casting study. The heaviest sections of this part are near the center of the flat wall (Figure 5.10a), and the next heaviest regions are the four bosses (Figure 5.10b). In all, five points were sampled from these heavy sections (see Figure 5.13). As can be seen from the cooling curves (Figure 5.14) and numerical simulation results (Figure 5.11 and 5.12), the heaviest region shows a good correspondence with the last region to solidify. In particular, when 95% of the casting has solidified, from Figure 5.13, we can see that the metal at PNT 0 (the heaviest section) has just starting to release its latent heat, whereas at 90% solidification, the second heaviest sections start solidifying.

(a) The Heaviest Region

(b) The Relatively Heavier Region
(Second Heaviest)

(c) Distance Distribution of the Object (A Section View)

Figure 5.10 Analysis Results of the Visualization System (Case 3)

Figure 5.11 Solidification Results for 95 % Solidification (Case 3)

Figure 5.12 Solidification Results for 90% Solidification (Case 3)

Figure 5.13 Location of Points Sampled from the Heaviest Section (Case 3)

Figure 5.14 Cooling Curves for the Sampled Points (Case 3)

5.2.4 Case 4

Figure 5.15 is another part obtained from a diecasting company that has a shrinkage porosity problem. Since the part has a very thick boss on one end, the gates are placed there. Even with this configuration, the thick boss still has the porosity problem. This creates many problems since a pressure tight hole must be machined through the boss. As can be seen from Figure 5.15, this problem can be easily identified by using the visualization system. In this case, a straightforward solution to this problem is to core out the heavy boss by using a moving slide in the die. Obviously, this will increase the die complexity. However, our system can at least identify the heavy section problem immediately which can then be used to facilitate the decision making.

Figure 5.16 shows the temperature profile of the numerical simulation from MAGMASoft. It can be seen from the two figures that the thickest section corresponds to the hottest regions. Figure 5.17 shows the location of the points being sampled from the heaviest section, and the corresponding cooling curves are shown in Figure 5.18. As indicated in Figure 5.18, when 95% of the total casting has solidified, the metal at those points is not yet starting to release its latent heat. Therefore, the heaviest sections are located in the regions where the last 5% of the metal solidifies.

(a) Opaque View of the Object

(b) The Relatively Heavier Region

(c) The Heaviest Region

(d) Distance Distribution of the Object
Near the Heavy Section (A Section View)

Figure 5.15 Analysis Results of the Visualization System (Case 4)

Figure 5.16 Numerical Simulation Results (Case 4)

Figure 5.17 Location of Points Sampled from the Heaviest Section (Case 4)

Figure 5.18 Cooling Curves for the Sampled Points (Case 4)

5.2.5 Case 5

Figure 5.19 shows another part that was sent to us by our member company. The wall thickness transition of this part, in general, is smooth. However, because of the heavy region caused by the transition near the two small bosses (see Figure 5.19), the part still has porosity problems that prevent pressure tightness, resulting in a high scrap rate. This heavy region problem near the transition regions can be possibly overlooked in the design stage. However, it can be easily identified by using the visualization system, as can be seen from Figure 5.19. The results can be used to modify the geometry by further smoothing the wall thickness transition.

Figure 5.20 shows the temperature profile of the numerical simulation from MAGMAsoft. It can be seen from the two figures that the thickest section corresponds to the hottest regions. Figure 5.21 shows the location of the points being sampled from the heaviest section, and the corresponding cooling curves are shown in Figure 5.22. As indicated in Figure 5.22, when 95% of the total casting has solidified, the metal at those points is still in the process of releasing its latent heat. Therefore, the heaviest sections are located in the regions where the last 5% of the metal solidifies.

(a) Opaque View of the Object

(b) Distance Distribution of the Object Near the Heavy Section
(A Section View)

(c) The Relatively Heavier Region

(d) The Heaviest Region

Figure 5.19 Analysis Results of the Visualization System (Case 5)

Figure 5.20 Numerical Simulation Results (Case 5)

(a) The Original 3-D object

(b) Object Being Sectioned Along the Heaviest Section

Figure 5.21 Location of Points Sampled from the Heaviest Section (Case 5)

Figure 5.22 Cooling Curves for the Sampled Points (Case 5)

5.2.6 Case 6

Figure 5.23 shows another die casting. The heaviest region was located near the four corners as shown in Figure 5.23. These heavy sections correspond quite well to the hot regions shown in the temperature profile in the MAGMASoft simulation (see Figure 5.24). For the quantitative comparison, eight points were sampled from these heavy sections (see Figure 5.25). As can be seen from the cooling curves (Figure 5.26) and numerical simulation results (Figure 5.24), when approximately 92% of the total casting has solidified, the metal at those points is still in the process of releasing its latent heat. Therefore, in this case, the heaviest sections are located in the regions where the last 8% of the metal solidifies.

(a) Opaque View of the Object

(b) Distance Distribution of the Object Near the Heavy Section
(A Section View)

(c) The Relatively Heavier Region

(d) The Heaviest Region

Figure 5.23 Analysis Results of the Visualization System (Case 6)

Figure 5.24 Numerical Simulation Results (Case 6)

Figure 5.25 Location of Points Sampled from the Heaviest Section (Case 6)

Figure 5.26 Cooling Curves for the Sampled Points (Case 6)

5.2.7 Case 7

Figure 5.27 shows the sectioned geometry of a die casting. This part has a high scrap rate due to the presence of a heavy region which prevented pressure tightness after machining. This problem was found after the part was in use in the field. In fact, manufacturing problems caused by this design forced the company to close down some assembly plants, causing delays and losses of millions of dollars. Upon using the visualization software on the same design the company was able to find the design defect immediately (they had found it by other means previously). Figures 5.27-5.30 shows the heaviest regions from the visualization system. Figures 5.31-5.35 show the temperature profile of the numerical simulation from MAGMAsoft. It can be seen from these figures that the thickest section corresponds to the hottest regions. The points being sampled from the heaviest sections are marked in Figure 5.30, and the corresponding cooling curves are shown in Figure 5.35. As indicated in Figure 5.36, when 95% of the total casting has solidified, the metal at those points (except Pnt 1) is still in the process of releasing its latent heat. Therefore, the majority of the heaviest sections are located in the regions where the last 5% of the metal solidifies.

In this case, the company has used pressure pins in the die and devoted much attention to address this porosity problem. However, the scrap rate is still high. They concluded that if they had had the tool available during the early stages of design, they would have found the defect early, when changes could have been made easily and inexpensively.

(a) Opaque View of the Object

(b) Semi-Transparent View of the Heaviest Section

Figure 5.27 Analysis Results of the Visualization System (Case 7: View 1)

(a) Opaque View of the Object

(b) Semi-Transparent View of the Heaviest Section

Figure 5.28 Analysis Results of the Visualization System (Case 7: View 2)

(a) Opaque View of the Object

(b) Semi-Transparent View of the Heaviest Section

Figure 5.29 Analysis Results of the Visualization System (Case 7: View 3)

(a) Opaque View of the Object

(b) Semi-Transparent View of the Heaviest Section

Figure 5.30 Analysis Results of the Visualization System (Case 7: View 4)

Figure 5.31 Numerical Simulation Results (Case 7)

Figure 5.32 Numerical Simulation Results (Case 7: Section 1)

Figure 5.33 Numerical Simulation Results (Case 7: Section 2)

Figure 5.34 Numerical Simulation Results (Case 7: Section 3)

Figure 5.35 Numerical Simulation Results (Case 7: Section 4)

Figure 5.36 Cooling Curves for the Sampled Points (Case 7)

5.2.8 Case 8

Figure 5.37a shows another part that was obtained from a member company that has a thermal-related manufacturing problem. Because this part must function under high pressure, the porosity near the joint regions (near the heavy sections shown in Figure 5.37b-5.37d) causes many of the parts to fail in the field. Again, the scrap rate is high and the manufacturing cost is unavoidably increased. This problem was easily identified by using the visualization system. Figure 5.38 shows the temperature profile of the numerical simulation from MAGMASoft. It can be seen from these figures that the thickest section corresponds to the hottest regions. Figure 5.39 shows the location of the points being sampled from the heaviest section, and the corresponding cooling curves are shown in Figure 5.40. As indicated in Figure 5.40, when 95% of the total casting has solidified, the metal at those points is still in the process of releasing its latent heat. Therefore, the heaviest sections are located in the regions where the last 5% of the metal solidifies.

(a) Opaque View of the Object

(b) The Heaviest Region

(c) Another View of the Heaviest Region

(d) Distance Distribution of the Object Near the Heavy Section
(A Section View)

Figure 5.37 Analysis Results of the Visualization System (Case 8)

Figure 5.38 Numerical Simulation Results (Case 8)

(a) The Original 3-D object

(b) Object Being Sectioned Along the Heaviest Section

Figure 5.39 Location of Points Sampled from the Heaviest Section (Case 8)

Figure 5.40 Cooling Curves for the Sampled Points (Case 8)

5.2.9 Re-Design Example

Figure 5.41(a) shows a preliminary design that was provided by a die casting company. This part has a thick section near the two mounting flanges. Figure 5.41(b)-(g) shows the analysis results from the visualization system. The heavy section is clearly visible from the pictures. In this case, we use qualitative wall thickness analysis (city-block metric) and each picture in the sequence is the result of peeling an additional four voxel layers. As can be seen from the sequence, the heavy section is much thicker than the rest of the part as shown by large number of steps for which the red core persists. Figures 5.42 and 5.43 show the temperature profile of the numerical simulation. Again, in these figures, the hottest regions are near the centers of the mounting ears. Figure 5.44 shows a CAD drawing with the two points sampled from the heaviest sections, and their cooling curves are shown in Figure 5.45. From the cooling curves, we find that

(a) when 95% of the total casting has solidified, the metal at those points is just beginning to release its latent heat.

(b) it takes a relatively long time before these points finish releasing the latent heat.

The heavy section could be cored out to reduce the heavy sections and ribs added to retain strength. Figure 5.46 shows a re-designed part and Figure 5.47 shows the progressively heavy sections of this re-designed part. The analysis is based on the same voxel resolution that was used for the original part, and qualitative analysis is used. As in the case of the original design, each picture shows the results of peeling an additional four layers from the object. As can be seen from this sequence of pictures, the wall thickness of the re-designed part is much more uniform. Figures 5.48 and 5.49 show the temperature profile of the re-designed part based on the numerical simulation when 95% of the part are solidified. In this case, the time required for solidification is about 50% of the time for the original part. Therefore, the cycle time can be shortened assuming there are no problems in the bisquit region.

(a) Original Design

(b) Heavy Section

(c) Heavy Section

Figure 5.41 Sequence of Progressively Heavy Sections Obtained From the Analysis Results of the Visualization System (Original Design)

(c) Heavy Section

(d) Heavy Section

(e) Heavy Section

(f) Heavy Section

Figure 5.41(cont'd.) Sequence of Progressively Heavy Sections Obtained From the Analysis Results of the Visualization System (Original Design)

Figure 5.42 Numerical Simulation Results (Original Design)

Figure 5.43 Sectioned View of the Numerical Simulation Results (Original Design)

Figure 5.44 Location of Points Sampled from the Heaviest Section (Original Design)

Figure 5.45 Cooling Curves for the Sampled Points (Original Design)

Figure 5.46 The Geometry of the Re-Designed Part

(a) Heavy Section

(b) Heavy Section

(c) Heavy Section

(d) Heavy Section

Figure 5.47 Sequence of Progressively Heavy Sections Obtained From the Analysis Results of the Visualization System (Re-Design)

Figure 5.48 The Numerical Simulation Results (Re-Design)

Figure 5.49 Sectioned View of the Numerical Simulation Results (Re-Design)

	volume	# of voxels	# of polys	voxelization	DT(qualitative)
Case 1	150*47*66	465300	242	4 sec.	4 sec.
Case 2	150*63*58	548100	1812	16 sec.	4 sec.
Case 3	150*76*143	1630200	328	15 sec.	12 sec.
Case 4	130*116*150	2262000	4940	72 sec.	18 sec.
Case 5	150*150*90	2025000	2840	40 sec.	15 sec.
Case 6	150*133*83	1655850	2470	58 sec.	13 sec.
Case 7	150*75*61	686250	9534	119 sec.	5 sec.
Case 8	150*110*130	2145000	3030	63 sec.	17 sec.
Re-Design(old)	140*57*150	1197000	12616	110 sec.	9 sec.
Re-Design(new)	139*59*150	1230150	13664	115 sec.	9 sec.

Table 5.1 The System Running Time for the Case Studies and Re-Design Examples

5.3 System Running Time

Table 5.1 lists the running time for the above 10 parts. These results are obtained by running this system on an IBM R6000 37T workstation with 128 MB memory. As can be seen, the visualization system takes at most 2-3 minutes to perform the analysis. On the other hand, the solidification simulation for a given part (Case 5) takes approximately 15 minutes to prepare the input model and about half an hour to perform the analysis. These do not include the post-processing time. To actually visualize the results, the user has to slice through the object and usually requires much manipulation of the display to construct an accurate mental model of the results.

5.4 Feedback From Industry

The system has been evaluated by members the NADCA computer modeling task group, and is in use at several NADCA member companies. They have provided valuable feedback regarding the experience they have with the system and have requested several extra functions. For example, besides checking the uniformity of the wall thickness, one company wants to use the tool to measure the wall thickness quantitatively to verify the wall thickness information obtained from their customer. Because of the inherited resolution errors involved with the voxel model and distance transform, a very high resolution volume is needed to fulfill this need. In fact, its use as a truly quantitative measure actually goes beyond the original intent of this research. Furthermore, this information can best be obtained from a B-Rep model where the measure can be exact. However, since some companies may not have the CAD model or even a CAD system, we have provided a object space cutting tool. By using this tool, they can perform the analysis on a user-specified local region with a finer resolution. In addition, this cutting tool can be used on big parts, e.g. transmission case. In this case, they can do a rough

analysis of the whole part based on a relatively coarse resolution and then use this cutting tool to perform a more thorough analysis for the regions with potential problems. The companies have also made many useful comments regarding the consistency of the user interface and have sent program bug reports. The reports have been used as a guideline to improve the usefulness of this system.

6. QUALITATIVE TECHNIQUES FOR FILL

Filling of the dies has a significant effect on part quality, and so, is one of the predominant physical phenomena of the die casting process. In spite of numerous simplifying assumptions, numerical modeling of metal flow still requires relatively long computation times. Furthermore, the use of flow modeling systems requires much experience and a deep understanding of fluid dynamics.

From a design perspective a great deal can be done with much simpler models. At the stage when design decisions are made, two types of information are useful in reducing flow problems. First, knowledge of the flow distances can help determine the direction from which fill should occur and whether any regions are too far from the gates. Second, an approximation to the fill pattern can further answer questions about the possibility of air entrapment, approximate locations for vents and facilitate comparisons between various gate locations.

6.1 *Specification of Gates*

The first step in evaluating the filling must be a way of specifying where that flow originates. Since the geometry of the part has a significant effect on how the part will fill, while choosing candidate locations for gates, overflows and vents, it is important to consider the effect the part geometry will have on the filling.

A simple means of specifying candidate gate locations was developed that allows the user to quickly define candidate gate locations. Evaluation can be based on single gates or multiple gates. The gate locations are specified by choosing points on the part surface that bound the candidate gates. The cross sectional area of these regions then represents the cross section of the gate or the region where the flow originates.

The surface points that bound the gates can be extracted from most commercial CAD systems¹. We have developed a tool to automatically extract them for one commercial CAD system, Pro/ENGINEER™ as shown in Figure 6.1. The coordinates of these points extracted must be relative to the same coordinate system used for generation of the SLA file. Since the coordinates of these points bounding the gate are passed to the filling evaluation tool through a file as shown in Figure 6.2.

The cross sections of the candidate gate boundaries are converted to triangular polygons whose vertices specify the boundaries. Since the cross sections of these boundaries are converted to polygons, when specifying gates on surfaces with high curvature, it is necessary to choose points close together so that the curvature of the surface is closely followed. There is no limitation on the number of points that can be used to specify a gate location.

The initial flow direction for the gates is assumed to be along the inward pointing normal vectors of the gate polygons. Liu (1997) developed a system that can place rudimentary fan gates on stereolithography models. An illustration of this is shown in Figure 6.3. The

¹ Later implementations of the system allow users to pick gates on the surface of the STL model eliminating the need for the use of a CAD system.

cross section of the runner of such a fan gate would then be used as the initial cross section, normal to which the initial flow direction is defined.

Figure 6.1 Tool for Specification of Gate Boundary Points Developed for Pro/ENGINEER™

To find the voxels that correspond to the gate, the gate polygons must be scan converted. The procedure used to scan convert surface polygons discussed in Chapter 2 is used to scan convert the gate polygons. The inward pointing normal to the gate polygon is associated with each gate polygon during voxelization. This defines the initial fill direction for that gate voxel. These gate voxels correspond to the gate locations that will be used in the flow distance calculations and the initial location from which the fill originates in fill pattern calculations.

Figure 6.2 Specification of Coordinates of Gate Boundary Points With a File

Figure 6.3 Example Part with a Rudimentary Fan Gate (Liu, 1997)

6.2 Distance from the Gate

Incorrect gate locations are responsible for incompletely filled cavities. When this occurs, usually the walls farthest from the gates are incompletely filled (Halliday, 1953). One of the factors that plays a significant role in proper filling of the die casting is flow distance (Cocks, Wall, 1983). Flow distance is an important factor in the determination of fill time, and a major determining factor in the elimination of defects (Bennett, 1961; Ramalingam, Zhao, 1987). Flow within the cavity should be such that no region takes much longer to fill than others (Barton, 1977). According to Barton (1973), a broad principle is to reduce the flow paths to a minimum.

The locations of the gates should ensure that the metal reaches the far periphery of the die cavity simultaneously so that portions of the casting do not solidify before the cavity is completely filled (Barton, 1973). Long flow distances increase the resistance to flow, reducing the rate of mass flow and increasing the heat loss to the die per unit volume of metal at the advancing front (Barton, 1963). Very long flow distances are responsible for bad surface finish in the regions farthest from the gate (Halliday, 1954).

Therefore, once the designer has identified candidate locations for the gates based on how the part may be oriented in the dies, flow distance is the first criterion that can be used to screen out unacceptable candidates. The flow distance for any region corresponds to the shortest possible distance the metal can travel to reach that region from the gate.

Measuring the flow distance involves finding the distance from any point in the casting to the gate, measured through the space occupied by the casting. Currently die casters estimate this distance using experience and visual inspection. This is a difficult task and the quality of the results depends greatly on the experience of the die caster.

Distance transform is a discrete distance measuring technique originally developed for image processing. Basically for each voxel in the object the distance transform computes the discrete distance from the "source" voxels. In this case the source voxels are the gate voxels identified during gate specification. Based on this, the regions that are relatively far from the gates, i.e. regions with long flow distances, can be identified as shown in Figure 6.4.

Linear distance is most commonly measured in terms of Euclidean distance. In a discrete world, like voxel space, an approximation to the continuous distance is used to measure distances from the "source" voxels to the other voxels of the part. The computation of this discrete metric of distance is performed locally, a small neighborhood at a time. The distance between any two points, therefore, is the sum of local distances along the shortest path between the two points.

The discrete path G_k between two points p_1 and p_n is the series of distinct points $p_1, p_2, \dots, p_n$ in discrete space such that every p_i and p_{i+1} are connected by the prime vectors v_i defined for that metric (Verwer, 1991). Each prime vector is associated with a local distance which when summed along the discrete path yields the discrete length for that path $L(G_k)$.

$$\text{If } G_k = \{ u_1, u_2, \dots, u_n \}, u_j \in \{ v_i \}$$

$$\text{then } L(G_k) = \sum_{i=1}^n |u_i|$$

The discrete distance between points p_1 and p_n , $D(p_1, p_n) = \min\{L(G_k)\}$

Figure 6.5 shows a 2-D example of a 3x3 neighborhood with the two types of prime vectors. Using these 2-D prime vectors 3 types of metrics can be defined, namely the city block ($a=1$, $b=\infty$), chess board ($a=1$, $b=1$) and chamfer ($a=3$, $b=4$) metrics. Each of these have 3-D equivalents. For flow distance evaluation only an ordinal measure of distance is required. Hence the city block distance, which is the most efficient to calculate, is sufficient. Since no diagonal steps are taken, the 2-D distance defined by the city block metric $D(p_1, p_n) = |x_n - x_1| + |y_n - y_1|$ where coordinates of point p_1 is (x_1, y_1) and coordinates of point p_n is (x_n, y_n) .

Figure 6.4 2-D Example of Flow Distance Calculations

To calculate the city block distances for any part, the gate voxels are first assigned a distance of zero. The distance value is incremented and the 6-connected neighbors of gate voxels that belong to the part are found and assigned that distance value. The distance value is incremented again and the 6-connected neighbors of the voxels with the previous distance value that have not yet been processed are assigned that distance value. The procedure is repeated until all the part voxels have been assigned a distance value. This amounts to the even distance voxels being further neighbors of the next lower valued-odd distance voxels and the odd distance voxels being further neighbors of the previous even distance voxels. This makes the use of dual stacks, one for the odd and one for the even distance voxels using modulo arithmetic a very efficient method to implement the city block distance calculations.

Figure 6.5 Prime Vectors for a 2-D Example of a 3x3 Neighborhood

6.3 Fill Pattern

The fill pattern will help make choices about where overflows and vents should be placed to reduce air entrapment.

6.3.1 Assumptions and Simplifications

Research concludes that the filling process in high pressure die casting is largely controlled by inertia. The fluid stream flows through the cavity without much change in direction until it impinges upon the cavity walls.

The generation of qualitative fill pattern is related only to the initial fill direction and the geometry of the part. When the flow hits an obstruction, the resulting fill directions are dependent on the local region of impact. Specifically, they are dependent on the relationship between the fill direction and the state of the voxels in that local region of impact.

Based on the aforementioned observations, the following assumptions have been made in the calculation of the qualitative fill pattern:

- a) The fill pattern does not change direction until an obstruction (cavity wall or previously filled region) is encountered.
- b) The fill direction and volumetric distribution after striking an obstruction are dependent on the neighborhood surrounding the region of impact.

To reduce the computational complexity, a simplification is made in the calculation of the filling pattern. There is no conservation of volume over all even though volume is distributed within a neighborhood. That is, if all the voxels within a neighborhood are occupied, the volume flowing into the neighborhood is not distributed elsewhere². Based on these assumptions and simplifications, the qualitative fill pattern can be quickly calculated even for complex parts.

6.3.2 Description of the method

To aid in the discussion of the flow pattern calculations, definitions of some symbols and notations are provided:

O_a represents the occupancy state of voxel a . Voxels have only two states, namely unfilled represented by 0 and filled represented by 1.

p_a represents the position vector of point a

P_{ab} represents that voxel a is the parent of voxel b , i.e. voxel b is filled directly by metal exiting voxel a

V_{ab} represents the volume fraction in voxel units of metal exiting parent a filling child voxel b within a local neighborhood and can take on real number values between 0 and 1

S_{ab} represents the step at which child voxel b is filled where a is the parent voxel and is a positive integer.

$N_{26}(a)$ denotes all the 26-neighbors (vertex, face and edge adjacent) of voxel a

F_a denotes the flow vector for voxel a . It has values (x,y,z) where $x \in \{0,\pm 1,\pm 2,\pm 3\}$, $y \in \{0,\pm 1,\pm 2,\pm 3\}$, $z \in \{0,\pm 1,\pm 2,\pm 3\}$, and $|x|+|y|+|z| \neq 0$

Since the fill originates from the gates, the initial conditions for the calculations involve fill conditions for the gate voxels. The gate voxels have no parent voxels and the volume filling the gate voxels is 1-voxel unit. Furthermore, the gate voxels are filled at the first ordinal step. Hence if g is a gate voxel:

voxel g has no parent, represented by P_{0g}

$V_{0g} = 1$

$S_{0g} = 1$, and

F_g = inward pointing normal vector of the gate polygon

² This assumption is removed in later implementations of the approach.

Voxels of the part that are not gate voxels can be filled in one of two ways. They can be filled by flow exiting only one voxel, i.e. they may have only one parent, or they can be filled by metal exiting more than one voxel, i.e. they have more than one parent as illustrated in Figure 6.6.

Let us first consider the case where voxels have a single parent. Flow out of the current voxel can either encounter an obstruction or not encounter an obstruction. For the case where no obstruction is encountered, let k be the current voxel.

If P_{jk} , P_{ij} and P_{hi} as shown in Figure 6.8, then

$$\begin{aligned} V_{jk} &= V_{ij} \\ S_{jk} &= S_{ij} + (S_{ij} - S_{hi}) \\ F_k &= F_j \end{aligned}$$

For the case where the metal exiting a voxel encounters an obstruction, the fill volume exiting the voxel is distributed to the voxels in the neighborhood. In the computation of qualitative filling pattern a $3 \times 3 \times 3$ neighborhood is used. Let voxel a be the current voxel from which the metal is exiting, let voxel b be the obstructing voxel, and let P_{ca} . An example of this is illustrated in Figure 6.7.

The volume fraction flowing into any neighborhood voxel is dependent on its position relative to the flow direction. A 2-D illustration of this is shown in Figure 6.9. The volume fraction is related to the dot product between the flow direction and the relative position of the neighbor voxel to the current voxel.

That is, for all $k \in S = \{N_{26}(a) - \{a, b, c\}\}$, S being the set of voxels in the neighborhood other than voxels a , b and c , the volume flowing into it is related to $F_a \cdot (p_a - p_k)$

However $F_a \cdot (p_a - p_k) \in \mathcal{R}$ since it can take on values from -1 to +1, and fill volume

$\in \mathcal{R}^+$ since volume must be a positive number. To map from $\mathcal{R}$ to $\mathcal{R}^+$ the negative dot product with the largest magnitude in the neighborhood is added to every dot product. Therefore, the positive volume component C_k is defined to be:

$$F_a \cdot (p_a - p_k) + \min(F_a \cdot (p_a - p_m)) \text{ where } m \in S.$$

Let $S' = \{v | v \in S \text{ and where } O_v = 0\}$, i.e. S' is the set of empty voxels in the neighborhood.

For $q, n \in S'$ and, P_{an} , P_{dc} and P_{ca} as shown in Figure 6.10.

$$\begin{aligned} V_{an} &= V_{ca} * (C_n / \sum_q C_q) \\ S_{an} &= S_{ca} + (S_{ca} - S_{dc}) * (1/V_{an}) \\ F_n &= p_n - p_c \end{aligned}$$

Voxel A filled only by metal exiting Voxel B

Voxel A filled by metal exiting both Voxel B and Voxel C

Figure 6.6 Illustration of Voxels with One and More Than One Parent

- Fill vector exiting voxel a
 ○ Current voxel a
 ● Flow obstruction voxel
 ● 26 neighbor of voxel a
 ● Parent voxel of voxel a

Figure 6.7 Illustration of Obstruction in Local Neighborhood

Figure 6.8 Dependency of fill for voxel k and its ancestors

Figure 6.9 2-D Illustration of Fill Volume Fraction Dependent on Dot Product of Fill Direction and Relative Voxel Position

Figure 6.10 Dependency of Fill for voxel n and its ancestors

As mentioned earlier, the second major case is when a voxel is filled by more than one parent. Let P_{un} and P_{wn} . Then the total fill volume entering voxel n is the sum of the volume fraction of n before flow from a second parent originated, the volume fraction

from u and the volume fraction from w. As a result, the voxel n will fill at an earlier step than it would have if filled by only voxel u or only voxel w.

Let t be the ordinal step at which the flow from a second parent is initiated and P_{eu} and P_{fw} . The step at which it fills therefore depends on both how long after step t it would have filled if filled only by voxel u and how long after t it would have filled if filled only by voxel w. Let $S_{(u,w)n}$ symbolize the step at which voxel n is filled when it is fed both by voxel u and voxel w.

$$\therefore S_{(u,w)n} = \frac{1}{\frac{1}{S_{un} - t} + \frac{1}{S_{wn} - t}} + t$$

The resulting fill vector F_n is the sum of the fill vectors from u and w sized by the volume fraction of u and w respectively flowing into n, i.e. the resulting fill direction is closer to that direction along which voxel n fills faster. Let $F_{n(u)}$ symbolize the fill vector of voxel n if it were fed only by voxel u and $F_{n(w)}$ symbolize the fill vector of voxel n if it were fed only by voxel w. Then:

$$F_n = \left\| \frac{F_{n(u)}}{S_{un} - S_{ou}} + \frac{F_{n(w)}}{S_{wn} - S_{ow}} \right\|$$

6.4 Implementation

A prototype of both the flow distance and filling pattern tools were implemented on a UNIX platform using the C programming language and Motif for the user interface. The prototype was integrated as part of the *CastView* system (Miller, et. al. 1997).

After scan conversion of the object, the first step involves identification of the gate voxels. For this purpose, the boundary points of each gate location are read from the gate file into a linked list of gates and are used to define the vertices of triangular polygons for the gate to which they belong. Since a linked list is used for the gates, there is no restriction on the number of gates. These triangular polygons define the cross section of the gate regions and are scan converted using the method described previously to obtain the gate voxels. The inward pointing normal vectors of these triangular polygons is used

as the fill direction assigned to the gate voxels. These gate voxels can be used for both the flow distance calculations and the filling pattern calculations.

A CFB, consisting of a 3-D integer array is used to store both the flow distance values of the voxels and the filling pattern values. The same array also represents the shape of the part by using positive values to represent the part and negative values to represent non-part. Hence the flow distance and filling pattern values are always positive since they only correspond to the part. Since an array structure is used, it is easy to find neighbors of voxels by stepping in x, y, and z. Figure 6.11 is a flow chart of steps involved in calculation of flow distance and filling pattern.

In the case of the flow distance, dual memory stacks are used to calculate the flow distance of voxels since they are the most efficient. Mechanisms for pushing and popping the addresses of voxels onto and from the stacks facilitate quick storage and retrieval. One of the stacks is used for storage of voxel addresses with odd distance values and the other is used for storage of voxel addresses with even distance values. First the gate voxels are loaded into the stack that corresponds to odd distance value voxels. The current distance value is set to 1. As each of these voxels is popped off the stack, the voxel's contents in the CFB is set to the current distance value and the addresses of its previously unprocessed neighbors are found and pushed onto the other stack. Once all the voxels in the first stack have been processed, the current distance value is incremented and the same procedure is repeated for the voxels in the other stack. The voxels are popped off one at a time, the voxel contents in the CFB are set to the current distance value and the addresses of its previously unprocessed neighbors are found and pushed onto the other stack. This is repeated until all the part voxels have been processed (indicated by both stacks being empty).

For the case of the filling pattern, an insertion queue is used to determine the order of filling of the voxels. The queue contains information about the address of the voxel in it, the filling direction for that voxel, the steps it takes to fill that voxel, and the parent of that voxel.

Figure 6.11 Flow chart of Steps Involved With Calculating Flow Distance and Flow Pattern

The current filling step is first set to 1, the gate voxels are put into the queue and the queue is advanced. Each voxel coming out of the queue when the queue is advanced is considered to be filled at that filling step. The CFB value of that voxel is therefore set to the current filling step value. For each current voxel (obtained by advancing the queue), its filling direction is used to determine what the next voxel to be filled by this voxel will be.

If the next voxel is empty, the filling direction and steps it will take to fill the voxel are the same as for the current voxel. The voxel is added to the queue based on the steps it will take to fill it. If the voxel is in the queue, it is being filled from at least one other direction. The resultant of the current filling direction and the queue filling direction of the voxel is calculated and the new step at which the voxel will fill is used to insert it into the queue.

If on the other hand the next voxel has previously been filled or is not a part voxel, the neighborhood must be found. The relative orientation of the voxels in the neighborhood of the obstruction is found and this is used to determine the steps it will take to fill the neighborhood voxels. The filling direction for these neighborhood voxels is then calculated. Again, consideration is given as to whether the voxel is currently in the queue (i.e. being filled from another direction). If so the resultant of the new filling direction and the queue filling direction of the voxel is calculated and the new step at which the voxel will fill is used to insert it into the queue. If not, the voxel is added to the queue based on the steps it will take to fill it.

Once all the current voxels have been processed, the current filling step is incremented and the queue is advanced to obtain the next set of current voxels. The procedure is repeated until no more voxels are present in the queue indicating that all the part voxels have been processed.

Once the flow distance or flow pattern results have been calculated, the volume visualization routines can be used to view them as shown in Figure 6.12 and Figure 6.13. The same routines used for thickness are used here. The distance from the gate or fill pattern sequence number is assigned to each voxel, depending on the analysis. To the visualization routines, it makes no difference. These are simply different data of the same type.

The user can select the results viewing threshold. The voxels with flow distance or filling pattern values either above or below this threshold will be displayed. This is done by making voxels with the chosen range of values opaque during the template-based ray casting. Using sliders on the interface, the user can change the viewing direction and threshold in real time and thus identify problem regions.

Figure 6.12 Example of flow distance results

Figure 6.13 Example of filling pattern results

7. VERIFICATION OF FILL ANALYSIS

7.1 Introduction

A prototype fill visualization was implemented using the "C" programming language and the Motif and Xwindows windowing system. It currently runs on a range of UNIX platforms including IBM Risc, HP, SGI, Sun, and Dec Alpha, and has been integrated as part of die casting evaluation software tool *CastView* at the Ohio State University.

As discussed earlier, if die casting designs have regions that are relatively far from the gates, this could result in filling problems such as premature solidification. In addition, if regions that are the last to fill are not adequately vented this could result in entrapped air. The filling tool is intended to help the designer and die casting engineer reduce filling problems caused by these by providing information about the flow distance and filling pattern in a timely fashion.

Industrial participants have provided with many die casting part designs in the form of actual parts, blue prints, Pro/ENGINEER™ models and STL files. These have been used to evaluate and demonstrate the flow distance and fill pattern results. The analysis results have been compared with historical production data about the filling-related defects, the results of numerical simulations, and short shots for a number of parts.

7.2 Case Studies

In the following case studies we show the analysis results for several parts supplied to us by our industrial collaborators. Where numerical simulation results were available we have presented the analysis results side-by-side with the numerical simulation for comparison. We have also used this method to facilitate comparison between the short shots and the analysis results. Furthermore, in cases where we were granted permission to mention the existence and locations of defects we have done so to facilitate comparison with the analysis results. Where we have been granted permission to include pictures of the manufacturing defects and short shots, we have done so.

7.2.1 Case 1

Figure 7.1 shows a casting supplied to us by an industrial collaborator that had filling and other problems. The primary location of poor fill was reported to be located in the large lug on the right hand side. Figure 7.2 illustrates that with respect to flow distance this region was relatively far from the gate. In fact it was one of the furthest regions from the gate as illustrated in Figure 7.3. Furthermore, the lug is relatively heavy, the filling would probably have improved if the part were rotated in the die and the gate placed much closer to the lug.

Figure 7.4, Figure 7.5, Figure 7.6 and Figure 7.7 present a side-by-side comparison of the filling pattern results and EKK's FEM simulation. Unlike the model used for filling pattern analysis, the EKK model has an overflow. As is apparent from these figures, the lug is also the last region to fill. Overflows and vents should therefore be placed in the region highlighted in Figure 7.8 in such a way as to avoid air being trapped in the region that could result in filling problems. Short shots provided to us by the company

confirmed that the lug region was the primary region that was the last to fill, as shown in Figure 7.9 and Figure 7.10.

For this part and a resolution of $200 \times 142 \times 75$ voxels, generating the voxel model from the STL file which had 54,380 polygons took about 10min 38sec. Including reading the STL file. Since the analysis is geometric-based, no process conditions were required. The fill pattern results took 4 min. 45 sec. to compute and the flow distance results took 4 sec. to compute. All the above times were measured using a stopwatch and the test was run using a basic IBM Risc/6000 3/T workstation with 128 MB RAM, 1 GB diskdrive, and a Power Gt4e Graphics Adapter.

Figure 7.1 Casting With Gate Region Supplied By an Industrial Collaborator

Figure 7.2 Flow Distance Results in Order of Increasing Distance for Example Part Obtained from Industry Showing Lug is Relatively Far From Gate

Figure 7.3 Highlighting the Furthest Regions From the Gate (Dark Region)

Figure 7.4 Comparison Between Early Stages of Fill Pattern (Left) and Numerical Simulation Results (Right)

Figure 7.5 Comparison Between Intermediate Stages of Fill Pattern (Left) and Numerical Simulation Results (Right)

Figure 7.6 Comparisons Between Later Stages of Fill Pattern (Left) and Numerical Simulation Results (Right)

Figure 7.7 Comparison Between Latter Stages of Fill Pattern (Left) and Numerical Simulation Results (Right)

Figure 7.8 Highlighting the Last Region to Fill (Dark Region)

Figure 7.9 Short Shot of Part Showing Unfilled Regions at Intermediate Stage

Figure 7.10 Short Shot of Part Showing Lug Has Still Not Been Completely Filled In Latter Stages

7.2.2 Case 2

Figure 7.11 shows another part that was supplied to us by one of our industrial collaborators. For this part, the gates were located in the regions highlighted in Figure 7.12. Figure 7.13 is a shaded image of the part in the same orientation as Figure 7.12 for comparison of the gate positions. Since we were provided no information about the actual shapes of the gate region except for their location and size, straight gates were used both for the filling pattern analysis and for the EKK numerical simulation. From Figure 7.14, Figure 7.15, and Figure 7.16 it is apparent that the furthest region from the gate is part of the butterfly flange near the right hand side. Figure 7.17, Figure 7.18, and Figure 7.19 present a side-by-side comparison of the filling pattern results and EKK's FEM simulation. As can be seen from these figures, the butterfly flange is one of the last regions to fill. The filling of the casting could probably be improved by finding a means of venting the region near the butterfly flange and reducing its distance from the gate. Historical data seems to corroborate this.

Figure 7.11 Another Part Provided by Industry

Figure 7.12 Semi-Transparent View Of Part Showing Locations of the Gates

Figure 7.13 Shaded Image View Of Part in Same Orientation For Reference Of Gate Locations

Figure 7.14 Flow Distance Results in Order of Increasing Distance for Example Part Obtained from Industry

Figure 7.15 Highlighting the Far Regions From the Gate (Dark Regions)

Figure 7.16 Highlighting the Furthest Regions From the Gate (Dark Regions)

Figure 7.17 Comparison Between Early Stages of Fill Pattern (Left) and Numerical Simulation Results (Right)

Figure 7.18 Comparison Between Intermediate Stages of Fill Pattern (Left) and Numerical Simulation Results (Right)

Figure 7.19 Comparison Between Unfilled Regions (Dark Regions) Near End of Fill Sequence (Left) and Numerical Simulation Results (Right)

7.2.3 Case 3

Figure 7.20 shows a part obtained from an industry source that had about a 10% scrap rate. This was a result of cracking in the horizontal boss at the right hand side during machining operations when the internal passage was drilled and threaded. Almost all the cracks appeared in the top portion of the boss as shown in Figure 7.20. The cause of this problem was determined to be a filling defect (Chen, 1997). The gates were located along the length at the bottom of the casting and the overflows were located along the bottom at the opposite side to the gates. No overflows were placed near the horizontal boss on the right. According to flow distance calculations the top of the horizontal boss on the right is relatively far from the gate and is in fact the furthest region, as shown in Figure 7.21 and Figure 7.22. Furthermore, the filling pattern in Figure 7.23 and Figure 7.24 predicts this region is one of the last to fill. Hence an overflow and vent should have been placed near

the region. Reducing the flow distance, placing an overflow nearby and adequately venting the region could possibly have eliminated the problem.

The fill pattern also shows a region at the other end being one of the last to fill. However, it is much closer than the region on the other side. In addition, since that region was not subjected to the same machining and hoop stresses as the horizontal boss on the right, no problems may have become apparent.

Figure 7.20 Industrial Part showing Failures (Cracks) in Boss on Right Hand Side that Occurred Due to Filling Defects in the Boss

Figure 7.21 Flow Distance Results in Order of Increasing Distance for Example Part with Fill Defect Showing Right Hand Boss is Relatively Far From the Gates

Figure 7.22 Highlighting the Furthest Regions From the Gate (Dark Region) showing the Right Hand Boss is Furthest Region From the Gates

Figure 7.23 Fill Pattern Results Indicate that One of the Last Regions To Fill are Near the Right Hand Boss

**Figure 7.24 Highlighting the Furthest Regions From the Gate (Dark Region)
Showing That a Overflow and Vent Should Have Been Placed Near the Right Hand
Boss**

7.2.4 Case 4

It is difficult to match the visualization patterns precisely with short shots but comparisons have been made with short shots for a zinc part provided to us by an industrial collaborator. Figure 7.25, Figure 7.26, and Figure 7.27 show that the fill pattern and the short shots match fairly well.

Figure 7.25 Comparison of Fill Pattern Results (Early Stages) With Short Shot Obtained from Industry

Figure 7.26 Comparison of Fill Pattern Results (Intermediate Stages) With Short Shot Obtained from Industry

Figure 7.27 Comparison of Fill Pattern Results (Latter Stages) With Short Shot Obtained from Industry

7.3 Water Model Studies for Model Verification

Because of the temperatures involved and the presence of molten aluminum, zinc or magnesium alloys, it is extremely difficult and costly to directly observe the process of part filling in the real die casting process. Water analog studies are one of the most practical methods to determine how closely a filling model represents the behavior of metal in the real world.

Some of the most extensive water analog studies were conducted by (Smith and Wallace, 1963) and (Stuhrke and Wallace, 1964; Stuhrke and Wallace, 1965). Their experiments studied the effects of different gate shapes and placements, and different shaped obstructions on the filling of rectangular flat plate cavities 6 in. long, 4 in. wide and 0.125 in. thick. Their objective was primarily to study how the different gate locations and shapes of the obstructions affected the filling of the rectangular cavity.

Lee, et. al. (1995) also ran water analog studies on rectangular flat plate cavities that were 4 mm and 2 mm. using straight fan gates with gate thicknesses of 0.15 cm, gate velocity of 5 m/s, and pressures of 0.6 MPa. Their objective was to study the effect of cavity thickness and gate velocity on the filling of rectangular plate cavities.

For this work, water models were used to verify the filling behavior at the neighborhood of impact with an obstruction. Since our objective was to verify the fill pattern as the flow strikes walls at different angles, the transparent dies were designed to be modular with an insert.

Figure 7.28 Prince Test Stand with Dies Mounted and Visitrak Controller (Left) used for Water Model Experiment

A Prince test stand illustrated in Figure 7.28 was used to conduct the water model experiments and a Visitrak control and data acquisition system (left) was used to control and monitor the operation of the test stand. The shot cylinder for the test stand had an internal diameter of 1.959 in. Ratio of shot cylinder diameter to hydraulic plunger diameter was about 0.8. The maximum accumulator pressure P_A was 1800 psi and hence:

$$\text{Effective Pressure } P_3 = P_A \cdot \frac{A_{\text{Shot Cylinder}}}{A_{\text{Plunger}}} = 1800 \cdot 0.64 = 1152 \text{ psi}$$

where :

$A_{\text{shot Cylinder}}$ is area of the shot cylinder

A_{Plunger} is area of the plunger

Therefore $1152 \text{ psi} < P_{3\text{max}} \leq 1800 \text{ psi}$

The maximum plunger velocity was calculated to be 107.3 in/sec. If we use Darcy's Equation:

$$P_3 = \frac{\nu \cdot V_g^2}{2g \cdot C_d^2}$$

where:

ν is density

V_g is gate velocity

g is gravity constant

C_d is the discharge coefficient

For water, specific weight = 0.0361 lb/in³, density = 0.0361 slug/in³, and discharge coefficient was assumed to be 0.5 based on NADCA recommendations. However:

$$V_g = \frac{V_{\text{Shot}} \cdot A_{\text{plunger}}}{G_g}$$

where:

G_g is the gate area

V_{Shot} is the shot velocity

V_g is the gate velocity

Therefore Darcy's equation becomes:

$$1152 = \frac{0.001123}{2 * 0.25} \left(\frac{V_{\text{Shot}} * 3.014}{G_a} \right)^2$$

As a result:

$$G_a = 0.0243 * V_{\text{Shot}}$$

Since V_{Shot} must be less than 107.3 in/sec, it was chosen to be 60 in/sec to keep well within machine capabilities. The desired gate area G_a was therefore calculated to be 0.12458 in².

Now to obtain atomization, NADCA recommends that:

$$\nu * G_t * (V_g)^{1.7} \geq 750$$

where:

ν is density

G_t is gate thickness

V_g is the gate velocity

To meet this requirement, G_t was chosen to be 0.04 in. As a result, gate length $G_L = 3.645$ in. Since NADCA recommends that the ratio of runner width to depth be approximately 3:1, runner width R_L was chosen to be 0.5 in. and since the runner must have the same cross section as the gate, runner thickness $R_t = 0.29$ in.

To best meet the objectives of the experiment, a modular design approach was chosen. The ability to change the angle at which the water impacts a wall was one of the requirements so the transparent dies were designed so that this could be done easily. As shown in Figure 7.29 the transparent dies were constructed out of 9 in. wide by 18 in. high 3 layer sandwich of acrylic sheet each of which was 1 in. thick. The two outer layers of the sandwich were left transparent and unchanged except for the shot hole and hole where the shot sleeve can be recessed on one of the sides. The center layer of the sandwich is designed to be of 4 pieces. The first had the runner from the shot hole to the gate region machined into it. The other three pieces, shown in Figure 7.30 and Figure 7.31 consist of the gate region, the cavity and a removable insert.

A fan gate was used for the experiment as shown in Figure 7.32. The dimensions of the gate and runner cross section, shown in Figure 7.33 and Figure 7.34 were based on the gate and runner dimensions calculated above. The cavity region, as shown in Figure 7.35, was designed in relation to the capacity of the Prince test stand. The cavity was designed to be as deep as the middle acrylic layer, i.e. 1 in., and its other dimensions are shown in Figure 7.36. The cavity was also designed to hold a cavity insert firmly so that the experiment could be conducted such that the water could strike the cavity wall at 90° or the angled insert. The cavity insert was designed to be removable during the course of the experiments and was designed to have a 45° angled wall, as shown in Figure 7.37. The dimensions of the insert, as shown in Figure 7.38, were designed so that it fit snugly in the cavity when inserted and allowed the flow to strike the wall at a 45° angle.

Furthermore, a 1 inch wide and 0.1 inch deep vent was cut in the insert and middle layer of the dies to prevent back pressure.

To give the dies enough strength to withstand the pressures and forces, 3/8 inch threaded rod was used to bolt the three layers together as shown in Figure 7.39. To distribute this force evenly, 1 1/2 inch wide steel strips that were 3/8 inches thick were used on both sides along the locations where the nuts on the threaded rod were located. Drawings of this arrangement are shown in Figure 7.40, Figure 7.41, and Figure 7.42. Furthermore, 3/8 inch steel plates were bolted on the top and bottom and left and right sides to support the acrylic dies from forces in those directions. The plates were bolted to each other by 1/2 inch threaded rod as shown in Figure 7.39, but no holes were made in the acrylic to accommodate these bolts, retaining the strength of the material.

Figure 7.29 Water Model Apparatus Used For Experiment

Figure 7.30 Center Layer Gate Region and Cavity

Figure 7.31 Gate Region and Insert Modules

Figure 7.32 Gate Region Module

Figure 7.33 Runner and Gate Transition Dimensions (Top View)

Figure 7.34 Runner and Gate Transition Dimensions (Profile View)

Figure 7.35 Cavity Module Design

Figure 7.36 Cavity Module Dimensions

Figure 7.37 Insert Module Design

Figure 7.38 Insert Module Dimensions

Figure 7.39 Assembled Water Model Dies Mounted On the Test Stand

Figure 7.40 Front View of the Transparent Dies Indicating Dimensions of Holes and Steel Plates and Strips Used for Clamping

Figure 7.41 Top View of the Transparent Dies Indicating Dimensions of Holes and Steel Plates and Strips Used for Clamping

Figure 7.42 Side View of the Transparent Dies Indicating Dimensions of Holes and Steel Plates and Strips Used for Clamping

To mount the dies at a height so that the shot hole was level with the shot sleeve as shown in Figure 7.43, a wooden base of silver oak was made and the bottom steel plate attached to it through the use of 1/2 inch bolts. To prevent the dies from moving during the shot stroke, 4 "tie rods" made of 3/8 inch threaded rod were bolted to the dies at one end and to the support at the opposite end of the shot sleeve at the other end as shown in Figure 7.43. To provide further support, a 3/8 inch steel plate attached to 4 inch wide angle iron was bolted to the test stand behind the dies such that it prevented them from moving during a shot as shown in Figure 7.39. To prevent the cavity, gate and runner from leaking between the layers of acrylic under the high pressures, a thin film of water proof lithium grease was used between the layers.

To duplicate the kinematic viscosity of molten aluminum, a 7% by weight glycerine solution in water was used as the fluid for the experiment (Duran, et. al., 1991). One of the limitations of this simulation technique is reported to be the considerable difference in the surface tension values between molten aluminum and the glycerine solution in water. To make the glycerine-water solution clearly visible to the high speed camera, 40 drops of green food coloring were added per liter of solution.

Figure 7.43 Far End of Shot Sleeve Bolted to Dies To Prevent Movement

Figure 7.44 Velocity vs. Position Trace of Shot from Visitrak Data Acquisition System

The total stroke length was 20 inches. The shot length was 18 inches. The shot sleeve fill percentage was approximately 50%. The end of shot velocity was set to be 60 inches/sec. The velocity vs position profile is shown in Figure 7.44. For future water modeling experiments it is recommended that the plunger tip be lubricated to prevent it from sticking.

Figure 7.45 Side of Transparent Dies on which High Speed Video Camera was Placed

Figure 7.46 Profile View of Filling Pattern Results for Water Model Cavity with Insert

The filling of the cavity was electronically captured using a Kodak Ektapro 4540 high speed video camera with 64 Megs of memory. The camera uses a CCD array to obtain 256x256 pixels per frame and has a base frame rate of 4500 frames/sec (highest speed at which you can get full resolution). Each pixel has 8 bits of intensity data. The camera was placed on the side of the dies shown in Figure 7.45. The camera was fitted with a 50 mm camera lens.

To improve the contrast and visibility of the filling, the dies were backlit (on the shot sleeve side) with a centrally located 500 watt photo flood light located 1 1/2 ft behind the dies. The high speed video camera was placed approximately 2 feet from the dies and

photographers black drape was draped over the dies and camera and the region between to eliminate ambient light and reflections off the acrylic dies. The camera was automatically triggered by a positional microswitch and recorded the filling of the cavity at 4500 frames/sec for 0.22 sec at each shot. After each shot was recorded, the camera allowed playback at various frame rates. The filling sequence was played back at 2 frames/sec and transferred at that frame rate to SuperVHS tape.

Figure 7.46 shows a profile view of the filling pattern results for the water model of the cavity with the insert. As can be seen, the filling pattern moves up the wall on the gate side and then the pattern striking the side walls from the fan gate and from the flow hitting the 45° angle wall causes the spread into the rest of the cavity. This corresponds to the results obtained from the high speed camera. Figure 7.47, Figure 7.48, and Figure 7.49 show a comparison between the filling pattern results with the cavity insert and the corresponding frames from the physical model high speed video camera results. Figure 7.50 shows the same for no cavity insert. As can be seen from the pictures there is a fairly good correspondence between the filling pattern results and the physical model.

Figure 7.47 Comparison Between Early Wire Insert Filling Pattern Results (Left) and Physical Model Results (Right)

Figure 7.48 Comparison Between Intermediate With Insert Filling Pattern Results (Left) and Physical Model Results (Right)

Figure 7.49 Comparison Between Lather With Insert Filling Pattern Results (Left) and Physical Model Results (Right)

Figure 7.50 Comparison Between Without Insert Filling Pattern Results (Left) and Physical Model Results (Right)

7.4 Summary and Conclusions

The verification studies performed and described above show that the fill visualization techniques developed in this research do in fact work and work quite well. The general pattern of fill corresponds well to the fronts calculated with numerical simulation for the set of parts tested. The numerical results obviously provide much more information, but the qualitative results do indicate the general pattern and they provide a good indication of the last regions to fill.

The water analog study indicates good general correspondence, but a detailed review of the qualitative results some problems with implementation of the qualitative reasoning methods. The pattern from the gate of flow from the gate does not correspond with the characteristics that are assumed in setting up the calculations. The depicted pattern breaks up to quickly and does not follow the wall of the casting. These are numerical problems that are with the implementation that are being addressed in the PC version of the visualization software that is being developed in subsequent work.

The work has shown the feasibility of qualitative reasoning for fill and has shown that there is tremendous potential for assisting with the design of better die castings. Run times, particularly when you consider setup time as well, are very short and will become even shorter with improved implementation. The speed of response and simple setup should result in a useful and important tool.

8. SUMMARY AND CONCLUSIONS

Novel and efficient design evaluation methods have been developed. The methods work directly on the part geometry, require very little setup, and function in a nearly automatic manner. Computation times are very short.

The visualization approach that was developed provides consistent and robust results based on the overall part geometry. It provides a fast, yet complete, evaluation of the part design in terms of manufacturing relevant geometric characteristics.

Volumetric part representations and their associated processing techniques, such as distance transformation and thinning, have proved to be very efficient to extract volume-related geometric characteristics. Volume visualization techniques present the results in a natural fashion.

The fill visualization techniques compare favorably with the locations of the last regions to fill as predicted by numerical simulations. The results also correspond well with limited short shots and historical information. This tool can be used during design to evaluate candidate gate locations, evaluate locations for vents and overflows and to quickly compare designs to find one or two viable ones for consideration.

The concepts and methods developed were demonstrated with a prototype system implemented in UNIX. The resulting system is very efficient in terms of computation time, and can evaluate designs regardless of their shape complexity. The system is very simple to use and the results are self-evident. Consequently, the approach can be used much earlier in the design process than other tools, resulting in significant reductions in cost and effort.

All in all, the volume based, or voxel based, evaluation procedures have been shown to have considerable promise. Computation is extremely fast and the qualitative results are better than expected. Both fill and thermal issues seem to be addressed quite effectively with this approach.

8.1 *List of Accomplishments*

The specific accomplishments of this research are briefly listed below.

8.1.1 **Design evaluation**

- Thick sections of the part, thin sections of the part, and thin die regions are efficiently and effectively identified from the part geometry based on a voxel model.
- Distance from the gate and an approximate fill pattern can be efficiently computed on the same voxel representation.
- The results are presented using a specifically designed visualization system.
- The usefulness of this system has been demonstrated through several case studies and examples. The results correlate well with observed casting defects and/or with numerical simulation predictions.

- The voxel-based model was identified as the best part representation scheme to support die castability evaluation. Voxel-based models have the unique advantage of being able to model not only the surface but also the interior of an object explicitly. The uniformity and regularity of the representation simplifies reasoning about complex geometries.

8.1.2 Thick/thin section

- A systematic approach for modeling and extracting wall thickness from a CAD data was developed. Both the thick/thin sections of the part and the thin regions of the die can be efficiently identified. This is based on the distance transform and 3-D thinning skeleton of a voxel model.
- A new 3-D thinning algorithm was developed to facilitate the identification of thin regions. An effective method to identify simple points and new surface point conditions for the thinning skeleton was discussed.
- The approach was verified by comparing thick section displays with numerical simulation results for several parts. The correspondence between last regions to solidify and heavy sections is extremely good.

8.1.3 Fill visualization

- Qualitative filling evaluation methods that can provide filling pattern results in a few minutes were developed.
- Methods to calculate and display the shortest distance to the gate from any point in the casting were developed and implemented.
- A method for the user to specify potential gate locations in the CAD model and then automatically transfer the information to the voxels corresponding to those gate locations was developed.
- A novel method to calculate the filling pattern for die castings based on the gate locations and initial filling directions was developed and implemented.
- A transparent die was designed and built, and water model studies were performed. Filling was videotaped with a high speed video camera at 4500 frames/sec. The qualitative fill pattern visualization results correspond quite well to the video.
- The filling pattern analyses of several parts were compared with the results from numerical simulations. The results were surprisingly good.
- Filling pattern results were compared favorably with short shots.

8.1.4 Visualization methods

- A three-layer scheme was developed to facilitate the display of selected regions related to thickness and/or fill. This scheme not only improves the quality of the image, but also increases the efficiency of the image renderer.
- For better rendering of the analysis, a new 3-D contextual shading algorithm that supports the shading of binary voxel models was developed.

- A new template-based space-leaping method was developed to speed up volume rendering. This method can handle cases where the metrics used for the distance transformation and for the rays used for rendering are different.

8.2 *Future Work*

The basic methods developed in this research are currently being implemented in PC-based system running under Windows or Windows NT. This work is being completed with funding from the American Metalcasting Consortium. Several of the errors and inefficiencies present in the UNIX prototype are being eliminated. Included in the improvements are a more efficient implementation of the fill pattern calculations and an interface to identify gate locations while operating within the evaluation system. There is no need to use a CAD system. Gate boundaries are selected directly on the STL representation while in the STL view mode.

Technical improvements that should be considered at some point in the future are described below.

The voxel model presents an inherent resolution limit, and all the corresponding computation depends heavily on the volume size selected. Consequently, one of the most common questions encountered is specification of the appropriate volume size for the problem at hand. Currently, the system requires that the user specify the volume resolution, sometimes resulting in confusion and unnecessary computation. For example, if the user wants a coarse analysis, using an extremely fine volume would not be beneficial. On the other hand, if the desired error is small, a coarse volume may not produce any reasonable results. In the future, the system should be designed so the system can guide the user to specify the resolution based on the required analysis. This could be further extended to the stage where the accuracy of the STL file is specified.

In addition, in contrast to the current uniform voxel model, the possibility of using multiple-level voxel models should be explored. For example, one resolution can be used for the part voxel and another for the die. In this way, a more accurate quantitative measure of wall thickness can be obtained without the severe memory requirements and processing time.

Extensions to the fill evaluation include a study of the effects of flow hitting obstructions at various velocities and modification of the obstruction neighborhood. This would permit use with other casting processes in addition to high pressure die casting. This type of extension is necessary to properly account for behavior in the runner.

It may be possible to associate some resistance with unfilled voxels to represent back pressure and resistance to fill as the cavity fills.

Future work also includes the development of techniques to combine heavy section information with the filling information to identify transitions in wall thickness along the fill path. This would help identify problems such as thin regions feeding heavier regions that prematurely solidify.

REFERENCES

- Adler, R.E. and Ishii, K., 1989, "DAISIE: Designer's Aid for Simultaneous Engineering," *ASME Computers In Engineering*, pp. 19-26.
- Ahuett-Garza, H., Miller, R.A., Kinzel, G.L. 1996. "Characterization of loads in die casting and prediction of die deflections", Ohio State University Engineering Research Center for Net Shape Manufacturing, ERC/NSM C-96-28.
- Andresen, W.T. 1987. "Computer simulation and analysis of liquid metal flow and thermal conditions in die casting dies", *SDCE 14th International Die Casting Congress and Expo*, G-T87-019, 1-5.
- Ayala, D. 1988. "Boolean operations between solids and surfaces by octrees: Models and algorithm", *Computer Aided Design*, v20, n8, 452-465.
- Backer, G.P. 1993. "Finite-element free surface flow analysis: a new tool for foundry engineers", *Modelling of Casting, Welding and Advanced Solidification Processes VI*, 405-412.
- Barton, H.K. 1963. "Some flow problems associated with multi-point injection", *Machinery* London, 960-965.
- Barton, H.K. 1973. "Die design for thin-walled die castings", *Machinery and Production Engineering*, vol. 123, no. 3166, 122-126.
- Barton, H.K. 1977. "Gating layout criteria for die casting dies", *Machinery and Production Engineering*, vol. 130, no. 3358, 402-407.
- Barton, H.K., 1981, "*Product Design for Die Casting*," The Society of Die Casting Engineers.
- Beiter, K.A. and Ishii, K., 1991, "Geometry-Based Index for Predicting Sink Mark in Plastic Parts," Technical Report, ERC/NSM-P-D-91-61, Engineering Research Center for Net Shape Manufacturing, The Ohio State University, Columbus, OH.
- Bennett, F.C. 1961. "Gating magnesium diecastings", *Foundry*, vol. 89, no. 12, 75-79.
- Blum, H., 1967, "A Transformation for Extracting New Descriptors of Shape," Wathen-Dunn, W. Ed., *Models for the Perception of Speech and Visual Form*, M.I.T. Press.
- Boon, J.P., 1991, "Statistical Mechanics and Hydrodynamics of Lattice Gas Automata: An Overview," *Physica D* pp. 3-8.
- Borgefors, G., 1984, "Distance Transformations in Arbitrary Dimensions," *Computer Vision, Graphics, and Image Processing*, Vol. 27, pp. 321-345.
- Borgefors, G., 1986, "Distance Transformations in Digital Images," *Computer Vision, Graphics, and Image Processing*, Vol. 34, pp. 344-371.
- Borgefors, G., 1991, Another Comment on "A Note on 'Distance Transformations in Digital Images,'" *Computer Vision, Graphics and Image Processing*, Vol. 54, No. 2, pp. 301-306.

- Brieger, E and Bonomi, E., 1991, "A Stochastic Cellular Automaton Simulation of the Non-Linear Diffusion Equation," *Physica D*, pp. 159-168.
- Brieger, E and Bonomi, E., 1992, "Cellular Automata - Lattice Gas Models for PDE's," *Computer Physics Communications*, pp. 47-60.
- Brown, S.G.R., and Spittle, J.A., 1991, "Computer Simulation of Heat Transfer in Die Casting Procedures," *16th International Die Casting Congress*.
- Brown, S.G.R., and Spittle, J.A., 1992, "Rule-Based Lattice Computer Models for Simulating Dendritic Growth," *Scripta Metallurgica et Materialia*, Vol. 27, pp. 1599-1603.
- Brown, S.G.R., and Spittle, J.A., 1993, "A Rapid Alternative to Solidification Modeling," *Modern Casting*, pp. 24-25.
- Campbell, J. 1995. "Review of computer simulation versus casting reality", *Modelling of Casting, Welding and Advanced Solidification Processes VII*, 907-913.
- Chan, T.K., Gargantini, I., Walsh, T.R.S. 1989. "Conversion and integration of boundary representations with octrees", *Graphics Interface*, 203-210.
- Chen, C.L. 1997. "Evaluation of aluminum die casting defects causing casting rejection during machining", Masters Thesis, The Ohio State University.
- Chen, L., Herman, G.T., Reynolds, R.A., and Udupa, J.K., 1985, "Surface Shading in the Cuberille Environment," *IEEE Computer Graphics and Application*, Vol. 5, 33-43.
- Chen, Y.M., Miller, R.A., and Lu, S.C., 1992, "Spatial Reasoning on Form Feature Interactions for Manufacturability Assessment," *ASME Computers in Engineering*, pp. 29-36.
- Chu, C.N. and Lee, J.M., 1993, "Skeleton-Based Design Analysis of Near Net-Shape Products," *Annals of CIRP*, Vol. 42, pp. 193-196.
- Chvorinov, N., 1940, "Theory of Solidification of Castings," *Die Giesserei*, Vol. 27, pp. 177-225.
- Cocks, D., 1983, "Die Cast Components: Aid to Efficient Design," Zinc Development Association, London.
- Cocks, D.L., Wall, A.J. 1983. "Thermal aspects of zinc die casting", *British Foundryman*, vol. 76, pt. 3, 29-34.
- Cohen, D., Kaufman, A. 1991. "Scan-conversion algorithms for linear and quadratic objects", *IEEE Computer Society Tutorial: Volume Visualization*, 280-301.
- Crawfis, R.A. 1997. "Scientific visualization", Course Notes, Department of Computer and Information Science, The Ohio State University.
- Cunningham, J.J. and Dixon, J.R., 1988, "Designing with Features: The Origin of Features," *ASME Computer In Engineering*, pp. 237-243.
- Danielsson, P.E., 1980, "Euclidean Distance Mapping," *Computer Vision, Graphics and Image Processing*, Vol. 14, pp. 227-248.

- Davies, E.R. and Plummer, A.P.N., 1981, "Thinning Algorithms : A Critique and A New Methodology," *Pattern Recognition*, Vol. 14, Nos. 1-6, pp. 53-63.
- Duran, M., Karni, Y., Brevick, J., Chu, Y.L., Altan, T. 1991. "Minimization of air entrapment in the shot sleeve of a die casting machine to reduce porosity", Ohio State University Engineering Research Center for Net Shape Manufacturing, ERC/NSM C-91-31.
- Ebert, D.S., and Parent, R.E., 1990, "Rendering and Animation, of Gaseous Phenomena by Combining Fast Volume and Scanline A-Buffer Techniques," *Computer Graphics*, Vol. 24, No. 4, pp. 357-366.
- Exa 1996a. "Digital physics: Technology Background", Exa Corporation, Web Document: <http://www.exa.com/>
- Exa 1996b. "Digital physics: Simulation of flow over a backward-facing step", Exa Corporation, Web Document: <http://www.exa.com/>
- Flemings, M.C. 1974. "Solidification Processing", McGraw-Hill.
- Foley, J.D., Van Dam, A., Feiner, S.K., and Hughes, J.F., 1990, *Computer Graphics: Principles and Practice*, Addison-Wesley Publishing Company, 2nd Ed.
- Foreman, J.W. 1989. "Gaining competitive advantage by using simultaneous engineering to integrate your engineering, design and manufacturing resources", *AUTOFACT '89 SME Technical Paper*, 715.
- Gabriele, G.A., 1991, "The Application of Design for Assembly Principles to the Design of Aerospace Structures," *ASME Computers In Engineering*, pp. 225-232.
- Gadh, R and Prinz, F.B., 1992, "Recognition of Geometric Forms Using the Differential Depth Filter," *Computer-Aided Design*, Vol. 24, No. 11, pp. 583-598.
- Gadh, R., Hall, M.A., Gursoz, E.L., and Prinz, F.B., 1989, "Knowledge Driven Manufacturability Analysis from Feature-Based Representations," *ASME Computer in Engineering*, pp. 21-34.
- Gong, W. and Bertrand, G., 1990, "A Simple Parallel 3D Thinning Algorithm," *IEEE International Conference on Pattern Recognition*, pp. 188-190.
- Gordon, D. and Reynolds, R.A., 1985, "Image Space Shading of 3-Dimensional Objects," *Computer Graphics and Image Processing*, Vol. 29, pp. 361-176.
- Halliday, W.M. 1953. "Trouble-shooting die casting problems", *Foundry*, March, 197-326.
- Halliday, W.M. 1954. "Eliminating defects in die castings", *Foundry*, vol. 82, no. 12, 84-.
- Heine, R.W., 1968, "Feeding Paths for Riser Casting," *AFS Transactions*, Vol. 76, pp. 463-469.
- Heine, R.W., Uicker, J.J., and Gantenbein, D., 1984, "Geometric Modeling of Mold Aggregate, Superheat. Edge Effects, Feeding Distances, Chills, and Solidification Macrostructure," *AES Transactions*, Vol. 92, pp. 135-150.
- Herman, E.A., 1979, *Die Casting Dies: Designing*, Society of Die Casting Engineers.

- Hjalmarsson, H., Sudhalkar, A., Gursoz, L., and Prinz, F., 1994, "Automated Model Building for Moldability Analysis," *ASME Concurrent Product Design Conference*, DE-Vol. 74, pp. 71-85.
- Hohne, K.H. and Bernstein, R., 1986, "Shading 3D-Images from CT Using Gray-Level Gradients," *IEEE Transactions on Medical Imaging*, Vol. MI-5, No. 1, pp. 45-47.
- Holman, J.P. Ed., 1992, *Heat Transfer*, McGraw-Hill.
- Ishii, K. 1990. "Advanced optimization in design and manufacturing: Introduction to life-cycle design optimization" Course Notes, Department of Mechanical Engineering, The Ohio State University.
- Ishii, K. and Miller, R.A., 1992, "Design Representation for Manufacturability Evaluation in CAD: Beyond Feature-Based Design," *ASME Computers In Engineering*, pp. 37-44.
- Issac, J., Reddy, G.P., and Sharma, G.K., 1985, "Numerical Simulation of Solidification of Castings in Metallic Molds," *AFS Transaction*, Vol. 93, No. 14, pp. 123-132.
- Jense, G.J., 1989, "Voxel-based Methods for CAD," *Computer-Aided Design*, Vol. 21, No. 8, pp. 528-533.
- Joshi, S., and Chang, T C., 1988, "Graph-Based Heuristics for Recognition of Machined Features from a 3D Solid Model," *Computer-Aided Design*, Vol. 20, No. 2, pp. 58-66.
- Kajiya, J.T. and Von Herzen, B.P., 1984, "Ray Tracing Volume Densities," *Computer Graphics*, Vol. 18, pp. 165-174.
- Kaufman, A. 1987. "Efficient algorithms for 3D scan-conversion of parametric curves, surfaces and volumes", *Computer Graphics*, vol. 21, n. 4, 171-179.
- Kaufman, A. and Shimony, E., 1986, "3D Scan-Conversion Algorithms for Voxel-Based Graphics," *Proceedings ACM Workshop on Interactive 3D Graphics*, pp. 45-75.
- Kaufman, A., 1990, *Volume Graphics*, Academic Press.
- Kela, A. 1988. "Explicit hierarchical tree representation: A powerful utility", *Proceedings ASME Computers in Engineering*, 15-19.
- Kela, A. 1989. "Hierarchical octree approximations for boundary representation-based geometric models" *CAD*, vol. 21, no. 6.
- Kong, T.Y., and Rosenfeld, A.A., 1989, "Digital Topology: Introduction and Survey," *Computer Vision, Graphics and Image Processing*, Vol. 48, pp. 357-393.
- Kotschi, R.M., and Loper Jr., C.R., 1974, "Design of T and X Sections for Castings," *AFS Transactions*, Vol. 82, pp. 535-542.
- Kotschi, R.M., and Loper Jr., C.R., 1976, "Effect of Chills and Cores on the Design of Junctions in Castings," *AFS Transactions*, Vol. 84, pp. 631-640.
- Kotschi, R.M., and Plutshack, L.A., 1981, "An Easy and Inexpensive Technique to Study the Solidification of Castings in Three Dimensions," *AES Transactions*, Vol. 89, pp. 601-610.

- Kuipers, B. 1990. "The use of qualitative simulation in support of model-based reasoning", *SPIE Applications of Artificial Intelligence VIII*, vol. 1293, 204-209.
- Lam, L., Lee, S.W., and Suen, C.Y., Sep. 1992, "Thinning Methodologies-A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 9, pp. 869-885.
- Lee, T.-C., Kashyap, R.L., and Chu, C. -N, 1994, "Building Skeleton Models via 3-D Medial Surface/Axis Thinning Algorithms," *CVGIP: Graphical Models and Image Processing*, Vol. 56, No. 6, pp. 462-478.
- Lee, W.B., Lu, H.Y., Lui, Y.B. 1995. "A computer simulation of the effect of wall thickness on the metal flow in diecasting dies", *Journal of Materials Processing Technology*, vol 52, 248-269.
- Levoy, M., 1990, "Efficient Ray Tracing of Volume Data," *ACM Transactions on Graphics*, Vol. 9, pp. 245-261.
- Lewis, R.W., Usmani, A.S., and Huang, H.C., 1991, "Casting Modeling by Finite Elements - Our Experience," *Modeling of Casting, Welding and Advanced Solidification Process V*, Rappaz et al. ed., The Minerals, Metals and Materials Society, pp. 3-15.
- Li, R. 1991. "An algorithm for building octree from boundary representation", *ASME Intelligent Design and Manufacturing for Prototyping*, vol. 50, 13-23.
- Lin, H.J., Tsai, H.L. 1993. "Numerical simulation of an integrated filling-solidification casting system", *Modelling of Casting, Welding and Advanced Solidification Processes VI*, 381-387.
- Lindenbaum, M., Samet, H. 1990. "A probabilistic analysis of hierarchical geometric data structures", *IEEE Press*, CH2898-5, 91-96.
- Liou, S.Y., and Miller, R.A., 1991, "Design for Die Casting," *International Journal of Computer Integrated Manufacturing*, Vol. 4, No. 2, pp. 83-96.
- Liu, J. 1997. "A system to view and manipulate stereolithography STL files", MS Thesis, The Ohio State University.
- Lobregt, S., Verbeek, P.W., and Groen, F.C.A., 1980, "Three-Dimensional Skeletonization: Principle and Algorithm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 1, pp. 75-77.
- Loper Jr., C.R., and Kotschi, R.M., 1975, "Design of Bosses and L Sections for Castings," *AFS Transactions*, Vol. 83, pp. 173-184.
- Loper, Jr., C.R., Copur, M., and Kotschi, R.M., 1986, "Analysis of the Solidification Order in Selected Casting Sections," *AES Transactions*, Vol. 94, pp. 529-536.
- Lu, S. C., A. B. Rebello, D. H. Cui, R. Yagel, R. A. Miller, G. L. Kinzel 1996a, "A Visualization Tool for Mechanical Design," *Proceedings of Visualization '96*, San Francisco.
- Lu, S. C., A. B. Rebello, D. H. Cui, R. Yagel, R. A. Miller, G. L. Kinzel 1996b, "A 3-D Contextual Shading Method for Visualization of Diecasting Defects,"

Proceedings of Visualization '96, San Francisco.

- Lu, S. C., A. B. Rebello, R. A. Miller, G. L. Kinzel, R. Yagel 1997, "A Simple Visualization Tool to Support Concurrent Engineering Design," *Computer-Aided Design*, Vol. 29, No. 10, pp. 727-735.
- Lu, S. C., A. Rebello, R. A. Miller, G. L. Kinzel, and R. Yagel 1995, "A Volume-Based Geometric Reasoning Approach for Diecastability Evaluation," pp. 65-74, *ASME Computers in Engineering*.
- Lu, S.C. 1996, "Volume-Based Geometric Reasoning and Visualization to Support Manufacturability Evaluation in Die Casting," Ph.D. dissertation, Mechanical Engineering, The Ohio State University.
- Lu, S.C., Rebello, A.B., Cui, D.H., Yagel, R., Miller, R.A., Kinzel, G.L., "A 3-D contextual shading method for visualization of diecasting defects", *Proceedings of IEEE Visualization '96*, 405-407.
- Lu, S.C., Rebello, A.B., Miller, R.A., Kinzel, G.L., Yagel, R. 1995. "A voxel-based geometric reasoning approach for diecastability evaluation", *ASME Computers in Engineering Conference*, 65-74.
- Luby, S.C., Dixon, J.R., and Simmons, M.K., 1986, "Designing with Features: Creating and Using a features Database for Evaluation of the Manufacturability of Castings," *ASME Computers in Engineering*, pp. 285-292.
- Ma, C.M., 1995, "A 3D Fully Parallel Thinning Algorithm for Generating Medial Faces," *Pattern Recognition Letters*, Vol. 16, pp. 83-87.
- Maguire, M.C., Baldwin, M.D., Atwood, C.L. 1995. "Fastcast: Integration and application of rapid prototyping and computational simulation to investment casting", *27th International SAMPE Technical Conference*, 235-244.
- Malandain, G. and Bertrand, G., 1992, "Fast Characterization of 3-D Simple Points," *IEEE International Conference on Pattern Recognition*, pp. 232-235.
- Marks, P., 1994, *Process Reengineering and the New Manufacturing Enterprise Wheel: 15 Processes for Competitive Advantage*, published by The Computer and Automated Systems Association of the Society of Manufacturing Engineers CASA / SME Technical Forum.
- Mason, W.M., 1992, "Art From Cellular Automata and Symmetrized Dot-Patterns," *Computer & Graphics*, Vol. 16, No. 4, pp. 439-441.
- Masuda, E., Itoh, I., Haraguchi, K. 1995. "Application of mold filling simulation to die casting processes", *Modelling of Casting, Welding and Advanced Solidification Processes VII*, 747-754.
- Mcauley, A., 1992, "Improve Quality, Think Simple," *Design Net*, pp. 21-26.
- Menges, G, Baur, M.E., Lessenich, V., and Schenzer, C., 1988, "Computer-Aided Plastic Parts Design for Injection Molding," *Advances in Polymer Technology*, Vol. 8, No. 4, pp. 355-365.
- Meserve, B.E., 1955, *Fundamental Concepts of Geometry*, Addison-Wesley.

- Miller, R. A. A. Rebello, D. Cui, Y. Ma, T. Sun, S. Lu 1997, "Verification of Design Visualization Software," pp. 55 – 58, *Transactions 19th International Die Casting Congress and Exposition*, North American Die Casting Association, Rosemont, Illinois..
- Miller, R. A. , A. Rebello 1997, "Visualization of fill related Defects in Die Casting," pp. 5-9, *Transactions 19th International Die Casting Congress and Exposition*, North American Die Casting Association, Rosemont, Illinois.
- Miller, R.A., Rebello, A.B., Cui, D., Ma, Y., Sun, T., Lu, S.C. 1997. "Verification of design visualization software", *Transactions of 19th International Die Casting Congress and Exposition*, NADCA, Rosemont, Illinois.
- Molvig, K. 1994. "Digital physics", *Machine Design*, December 12, 1994.
- Montanari, U., 1968, "A Method for Obtaining Skeletons Using A Quasi-Euclidean Distance," *Journal of Association for Computing Machinery*, Vol. 15, No. 4, pp. 600-624.
- Morgenthaler, D.G., 1981, Three-Dimensional Simple Points: Serial Erosion, Parallel Thinning, and Skeletonization," TR-1005, Computer Vision Laboratory, Computer Science Center, University of Maryland, College park.
- Mortenson, M.E., 1985, *Geometric Modeling*, John Wiley & Sons.
- NADCA 1992. "Die casting defects", North American Die Casting Association, E-515.
- NADCA 1996a. "Gating die casting dies", NADCA Gating Die Casting Dies, Class Handouts: A, EC-514.
- NADCA 1996b. "Gating die casting dies", NADCA Gating Die Casting Dies, Class Text, EC-514.
- NADCA DDC, 1988, "*Product Design for Die Casting*," published by Diecasting Development Council of the North American Die Casting Association.
- Neises, S.J., Uicker, J.J., and Heine, R.W., 1987, "Geometric Modeling of Directional Solidification Based on Section Modulus," *AFS Transactions*, Vol.95, pp. 25-30.
- Ohnaka, I. 1993. "Modeling of fluid and solidification in casting", *Modelling of Casting, Welding and Advanced Solidification Processes VII*, 337-348.
- Osborne, M., Mobley, C., Miller, R.A., and Kallien, L., 1993, "Modeling of Die Casting Processes Using Magmasoft," *NADCA Transaction*, pp. 77-84.
- Ouyang, Q. 1994, "An Algorithm for Generating a Voxel-based Model from a B-rep Model," MS Thesis, Industrial and Systems Engineering, The Ohio State University.
- Packard, N.H. and Wolfram, S., 1985, "Two-Dimensional Cellular Automata," *Journal of Statistical Physics*, Vol. 38.
- Packard, N.H., 1986, "Lattice Models for Solidification and Aggregation," *Theory and Applications of Cellular Automata*, World Scientific, Singapore, pp. 305-310.
- Padhy, S.K., Sharan, R., Dwivedi, S.N., 1990, "Feature Based Approach for Casting

- Design," *Design Theory and Methodology: presented at ASME Design Technical Conferences*, pp 113-118.
- Patrikalakis, N.M. and Gursoy, H.N., 1990, "Shape Interrogation by Medial Axis Transform," *Advances in Design Automation: ASME Design Technical Conference*, pp. 77-88.
- Pavlidis, T., 1982, "Algorithms for Graphics and Image Processing," *Computer Science Press*.
- Perucchio, R., Saxena, M., Kela, A. 1989. "Automatic mesh generation from solid models based on recursive spatial decompositions", *International Journal for Numerical Methods in Engineering*, vol. 28, 2469-2501.
- Poli, C. and Shanmugasundaram, S., 1991, "Design for Die Casting: A Group Technology Based Approach," *Design Theory and Methodology: presented at ASME Design Technical Conferences*, Sep., pp. 135-141.
- Poli, C., and Fredette, L., 1989, "Product Design for Economical Die Casting," *The 15th International Die Casting Congress*, St. Louis, MO, Paper No. G-T89-074.
- Poli, C., Dastidar, P., and Mahajan, P., 1991, "Design for Stamping: Part I -Analysis of Part Attributes that Impact Die Construction Costs for Metal Stamping," *ASME Advances in Design Automation Conference*, DE-Vol. 32-1, pp. 33-41.
- Pratt, M.J., 1984, "Solid Modeling and the Interface between Design and Manufacturing," *IEEE CG&A*, Vol. 4, No. 7, pp. 52-59.
- Pratt, S.D., Sivakumar, M, and Manoochehri, S., 1993, "A Knowledge-Based Engineering System for the Design of Injection Molded Plastic Parts," *ASME Advances in Design Automation Conference*, DE-Vol. 65-1, pp. 287-295.
- Prinz, F. and Chern, J.H., 1988, "Geometric Abstractions Using Medial Axis Transformation," Technical Report, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA.
- Pro/ENGINEER User Guides, 1994, Release 14, Parametric Technology Corporation.
- R. Yagel, S.C. Lu, A.B. Rebello, and R.A. Miller 1995, "Volume-Based Reasoning and Visualization of Diecastability", *Proceedings of Visualization '95*, Atlanta, GA, pp. 359-362.
- Ramalingam, S., Zhao, J. 1987. "A unified scheme for the design of flow passages in die casting dies", *SDCE 14th International Die Casting Congress and Expo*, G-T87-017, 1-7.
- Ranjan V., 1991, "Thinning of 3-D Binary Patterns," Master Thesis, University of Calgary.
- Rebello, A. B. 1997, "Visualization of the Filling of Die Casting Dies," Ph.D. dissertation, Industrial and Systems Engineering, The Ohio State University.
- Rebello, A.B., Miller, R.A. 1993. "Alternate part representation to support design for die casting", Ohio State University Engineering Research Center for Net Shape Manufacturing, ERC/NSM D-95-15.

- Rebello, A.B., Miller, R.A. 1995. "Integration of physical phenomena evaluation with CAD systems", Ohio State University Engineering Research Center for Net Shape Manufacturing, ERC/NSM D-95-15.
- Requicha, A.A.G., Voelker, H.B., 1982, "Solid Modeling: A Historical Summary and Contemporary Assessment," *IEEE CG&A*, Vol. 2, No. 2, pp. 9-24.
- Rosen, D.W., Dixon, J.R., Dong, X. 1991. "A methodology for conversions of feature-based representations", *ASME Design Theory and Methodology*, vol. 31, 45-51.
- Rosen, D.W., Dixon, J.R., Poli, C., and Dong, X., 1992, "Features and Algorithms for Tooling Cost Evaluation in Injection Molding and Die Casting," *ASME Computers in Engineering*, pp. 45- 52.
- Rosenfeld, A. and Pfaltz, J., 1966, "Sequential Operations in Digital Picture Processing," *Journal of the Association for Computing Machinery*, Vol. 13, pp. 471-494.
- Rosenfeld, A. and Pfaltz, J., 1968, "Distance Functions on Digital Pictures," *Pattern Recognition*, Vol. 1, No. 1, pp. 33-61.
- Saha, P.K. and Chaudhuri, B.B., 1994, "Detection of 3-D Simple Points for Topology Preserving Transformations with Application to Thinning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 10, pp. 1028-1032.
- Salomons, O.W., van Houten, F.J.A.M., Kals, H.J.J. 1993. "Review of research in feature-based design", *Journal of Manufacturing Systems*, vol. 12, no. 2, 113-132.
- Sant, F., Backer, G. 1995. "Application of WRAFTS fluid flow modeling software to the bench mark test casting", *Modelling of Casting, Welding and Advanced Solidification Processes VII*, 983-990.
- Shah, J.J. and Rogers, M.T., 1988, "Feature Based Modeling Shell: Design and Implementation," *ASME Computers in Engineering*,.
- Shah, J.J., 1990, "The Design of Design Environments," *ASME Computers in Engineering*, pp. 281-288.
- Shah, J.J., 1991, "Assessment of Features Technology," *Computer-Aided Design*, Vol. 23, No. 5, pp. 331-343.
- Shih, T.S., and Li, M.Y., 1990, "Geometric Modeling of Casting Solidification," *Journal of the Chinese Society of Mechanical Engineers*, Vol. 11, No. 1, pp. 74-84.
- Sirrell, B., Holliday, M., Campbell, J. 1995. "The benchmark test 1995", *Modelling of Casting, Welding and Advanced Solidification Processes VII*, 915-933.
- Smith, W.E., Wallace, J.F. 1963. "Gating of die castings", *AFS Transactions*, v71, 325-343.
- Spur, G., Krause, F.L., Li, R., Lenz, E., Shpitalni, M., Fischer, A. 1990. "Multirepresentation-based geometrical modeler", *Annals of the CIRP*, vol. 39, n. 1.
- Srihari, S.N., Upuda, J.K., and Yau, M., 1979, "Understanding the Bin of Parts," *International Conference on Decision Control*, pp. 44-49.

- Stuhrke, W.F., Wallace, J.F. 1965. "Gating of die castings", *AFS Transactions*, v73, 569-595.
- Subramaniam, B.L. and Ulrich, K.T., 1994, "Producibility Analysis Using Metrics Based on Physical Models," *Design Theory and Methodology: presented at ASME Design Technical Conferences*, DE-Vol. 68, pp. 353-369.
- Sunderland, J.E. and Johnson, K.R., 1964, "Shape Factors for Heat Conduction Through Bodies with Isothermal or Convective Boundary Conditions," *ASHRAE Transactions*, pp. 237-241.
- Tsao, Y.F., and Fu, K.S., 1981a, "Parallel Thinning Operations For Digital Binary Images," *IEEE Conference on Pattern Recognition and Image Processing*, pp. 150-155.
- Tsao, Y.F., and Fu, K.S., 1981b, "A Parallel Thinning Algorithm for 3-D Pictures," *Computer Graphics and Image Processing*, Vol. 17, pp. 315-331.
- Tsao, Y.F., and Fu, K.S., 1982, "A 3D parallel Skeletonwise Thinning Algorithm," *IEEE Conference on Pattern Recognition and Image Processing*, pp. 678-683.
- Unruh, V.W., 1992, "A Framework for Multiple Domain Feature Modeling," Thesis, Purdue University.
- Vaghul, M.V., Zinsmeister, G.E., Dixon, J.R., and Simmons, M.K., 1985, "Expert Systems in a CAD Environment: Injection Molding Part Designs as an Example," *ASME Computers in Engineering*.
- Venkatesan, K. 1995. "Experimental and numerical investigation of the effect of process parameters on the erosive wear of die casting dies", Ph. D. Dissertation, The Ohio State University.
- Venkatesan, K., Shivpuri, R. 1995. "A Flow-3D investigation of the effect of gate geometry, gate velocity and gate size on the quality of die casting parts", Ohio State University Engineering Center for Net Shape Manufacturing, ERC/NSM C-95-18.
- Verwer, B.J.H., 1991, Local Distances for Distance Transformations in Two and Three Dimensions," Thesis, Delft University of Technology, pp. 23-33.
- Verwer, B.J.H., Verbeek, P.W., and Dekker, S.T., 1989, "An Efficient Uniform Algorithm Applied to Distance Transforms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, pp. 425-429.
- Viswanath, R. and Jaluria, Y., 1991, "Knowledge-Based System for the Computer Aided Design of Ingot Casting Processes," *Engineering with Computers*, Vol. 7, pp. 109-120.
- Vossepoel, A.M., 1988, A Note on "Distance Transformations in Digital Images," *Computer Vision, Graphics and Image Processing*, Vol. 43, pp. 88-97.
- Webber, R.E., 1990, "Ray Tracing Voxel Based Data via Biquadratic Local Surface Interpolation," *The Visual Computer*, Vol. 6, pp. 8-15.
- Woo, T.C., 1982, "Feature Extraction by Volume Decomposition," *Conference on*

- CAD/CAM Technology in Mechanical Engineering*, Cambridge, MA., pp. 76-94.
- Yagel, R. and Ciula, K. 1994, "High Quality Template-Based Volume Rendering", Technical Report OSU-CISRC-3/94-TR17, The Ohio State University.
- Yagel, R., and Kaufman, A., 1992, "Template-Based Volume Viewing," *Computer Graphics Forum*, Vol. 11, No. 3, pp. 153-157.
- Yagel, R., Cohen, D., and Kaufman, A., 1992, "Normal Estimation in 3D Discrete Space," *The Visual Computer*, 8, 5-6, pp. 278-291.
- You, I.C., Chu, C.N., and Kashyap, R.L., 1989, "Expert System for Castability Evaluation: Using a Fixed-Features Based Design Approach," *Robotics and Computer-Integrated Manufacturing*, Vol. 6, No. 3, pp. 181-189.
- Yueh, Y.S., Rebello, A.B., Miller, R.A. 1992. "Diecast: design for die casting", Ohio State University Engineering Center for Net Shape Manufacturing , ERC/NSM-D-C-92-65.
- Zhang, Y.F., Wang, H.P., Liu, W.K. 1994. "Fast-acting simulation of simultaneous filling and solidification", *Transport Phenomena in Solidification ASME Heat Transfer Division*, vol 284, 215-226.
- Zuiderveld, K. Z., Koning, A. H. J., and Viergever, M. A., 1992, "Acceleration of Ray Casting using 3D Distance Transform," *Proceedings of Visualization in Biomedical Computing*, R. A. Robb ed., Chapel Hill, NC, SPIE, Vol. 1808, pp. 324-335.
- Zuiderveld, K.J., Koning, A.H.J., Viergever, M.A. 1994. "Efficient volume traversal using distance transforms", Submitted to *IEEE Transactions on Visualization and Computer Graphics*