Proceedings of IDETC/CIE 2007
ASME 2007 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
September 4-7, 2007, Las Vegas, NV, USA

DETC2007-35165

A 3D Graphical Method for Interactively Determining the Vertical Convexity of Designs to be Molded

Mike Bailey
Oregon State University
mjb@engr.oregonstate.edu

Dru Clark
Scripps Institution of Oceanography
dclark@ucsd.edu

John Rapp
Microsoft Corporation
jrapp@microsoft.com

ABSTRACT

The Center for 3D Visualization Hardcopy (C3VH) was initiated in 1995 as a web-accessible facility to create 3D physical models. By making such prototyping hardware easy to access and the geometry files easy to check and correct, we have been able to drive the model-making process into the less traditional manufacturing area of scientific visualization. But, there have been requests for many copies of a single model. As "rapid" prototyping is actually quite slow and unsuitable for mass production, these requests have required us to enter the world of molding for mass production. As part of this, we have needed to quickly determine how feasible it is to mold a particular model. This paper shows an interactive OpenGL method that we developed for determining the optimal orientation for a model's parting line. By using a novel OpenGL stencil buffer technique, the user can re-orient the model on the screen, and see from the colors in the display how vertically convex it is in that orientation, and in what parts of the model it is not vertically convex.

INTRODUCTION

Within the C3VH project, areas such as earth science, biochemistry, geology, mathematics, medicine, aerodynamics, and climate studies are all finding new and insightful uses for these solid hardcopies. The most demanding of these visualization areas has been *biochemistry*, in particular the fabrication of complex protein molecules. From the protein kinase (Figure 1), to fasiculin, to the Light Harvesting II complex (Figure 2), biochemists have developed new insights into their molecules by having physical models that they can hold in their hands.

Figure 1: Protein Kinase Solid Model

In [HART96], Dr. Alexandra Newton, UCSD Professor of Pharmacology, talked about her work in crystal structures. Her research group had been studying the protein kinase C enzyme. Using the physical models from the C3VH, they discovered that a particular behavior of the enzyme was caused by part of the molecule covering up another key part. She said:

> "We had never thought about it in those terms before ... We knew it was not accessible, but we didn't know what was making it inaccessible. ... It was a revelation. ... We were surprised how much we learned from the solid model of the protein we were working with. Looking at it on the computer screen is just not the same as holding it and seeing all the nooks and crannies."

It is not surprising, then, that the demand to fabricate more molecules in the C3VH has been high. But, the process to fabricate a single complex

protein in the C3VH can take up to a day or two. So, the demand for solid molecule models greatly exceeds what can be produced.

As a quick aside, it is interesting to consider why these models have been so successful as 3D visualization display devices. We are currently planning a project to perform some rigorous scientific experiments to find out, but our qualitative observations have been these:

- The physical models appeal to our sense of touch. The tactile feedback from moving our fingers across a surface reinforces and enhances what the eye is seeing.

- The physical models allow us to visualize spatial relationships that are not visible on the graphics screen. Some researchers have been surprised when they pinch the center of their complex models and found how thick or thin they are.

- The physical models appeal to our sense of focus. The act of having to change our focus (accommodation) from the front of the object to the back gives us a sense of depth in a way that no flat computer graphics screen ever can. Even stereographics and virtual reality cannot achieve this.

- The physical models provide real-time collision information. This is probably the number one reason that this technology has touched such a nerve in chemistry and biochemistry. When attempting to understand the docking behavior of two molecules, there are so many contact surface comparisons that need to be made that complete numeric intersection tests cannot be done in interactive time. And, even if they could, they would still not be able provide the force and torque feedback to the hands and arms that real models can.

The Need for Identical Copies of the Same Molecule

The demand to fabricate molecules is quite high. But, in most cases, we have received requests, not for many different molecules, but for *many copies of the same molecule*. At first we found this surprising, but it turns out that there are two good reasons for this:

1. It is good to put more of the same molecule in the hands of multiple collaborators. In this way, many distributed collaborators can develop their own insights and compare them.

2. Some molecules, such as complexes and viruses, consist of a number of identical proteins arranged in a symmetric pattern. In order to fully understand how their docking geometry controls their behavior, they all need to be fabricated and assembled. An example of this is Dr. Klaus Schulten's Light Harvesting II complex, shown in Figure 2. Eight identical proteins were fabricated at the C3VH. Dr. Schulten's research group then hand-painted and assembled them. (See [SCHULTEN97] for more details.) Since then, we have gone through the same process with Schulten's sixteen-fold Light Harvesting I complex.

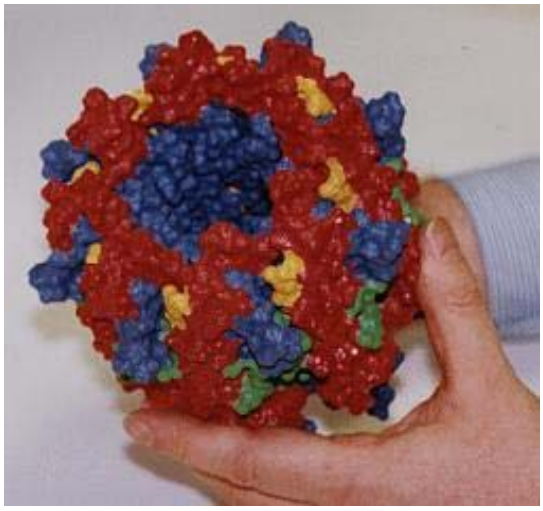


Figure 2: Light Harvesting II Complex Consisting of Eight Identical Protein Models

We have also been working with Dr. Jack Johnson at the Scripps Research Institute. Dr. Johnson's work is with viruses. We have made a number of identical proteins with him that have given him the tactile feedback to discover new information about how these pieces dock to replicate the virus. Our ultimate goal is to fabricate all *60* identical proteins needed to assemble the whole black beetle virus.

Thus, because of the usefulness of this technique, because of the demand, and because of the extensive time to fabricate a single molecule, we must now look to faster fabrication methods.

PREVIOUS WORK

The solution is to use mass production to fabricate more models in less time. So, we turn to industry to benefit from their experience. Industry attacks the mass production problem with a variety of techniques. One of the most common is numerically controlled (NC) machining. In NC machining, a tool bit is directed, under computer control, to carve the part from a solid block. But, NC machining will not work well here. The surfaces of the molecules are too complex. The tool bit would not be able to get into all of the nooks and crevices without accidentally gouging a piece of the part we wanted to keep.

Two other popular industrial techniques are casting and molding. Our overall problem is addressed in [FARAG89]:

> "...casting [and molding are] ... particularly suited for parts which contain internal cavities that are inaccessible, too complex, or too large to be easily produced by machining."

There are many aspects to molding processes, most aimed at production runs in the thousands to millions of piece parts ([MENGES93]). Here, we are targeting dozens of pieces, so we use a less formal process than would high-volume industry.

In a typical moderate-volume molding process ([HALLUM97], [PTC96], [PTE96], [WARNER96]), a positive of the model to be mass-produced is placed on a stand in a box, as shown in Figure 3:

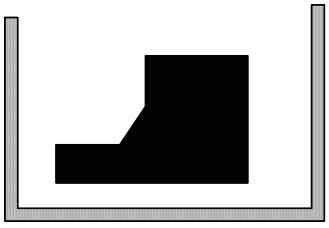


Figure 3: The Mold Process is started With a Part Positive

A slurry of mold-making material is poured in around the part up to a certain level and allowed to harden, or *cure*. The top of this surface is then lubricated with a compound called *mold release*. Then more mold making material is poured in to a level higher than the part. It is allowed to cure. This is shown in Figure 4.

Once hardened, the two halves of the mold are separated and the original part is removed. Access holes are then cut from the exterior of the mold to the interior. The hole created to allow the molten part material to get into the cavity is known as the *gate*. The one or more holes created to allow air and part material to get out are called *vents*. This is shown in Figure 5. The tunnels leading from the outside to the part interior are called the *runners*.

The two halves of the mold are then tightly bound together, usually in a custom-built box. Part material (e.g., plastic, wax, rubber, plaster) is injected into the gate until it is seen oozing out through the vents.

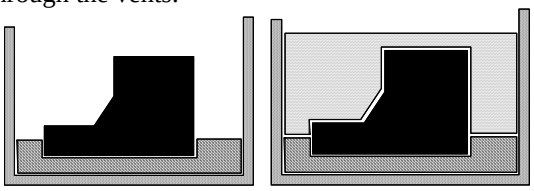


Figure 4: Mold Material Poured Around the Part

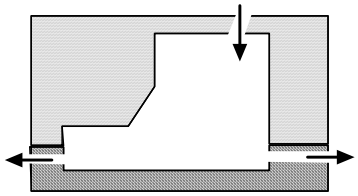


Figure 5: A Gate and Vents are Cut into the Mold Cavity

We have taken some of the molecules produced at the C3VH to low-volume industrial prototype shops and asked to have molded copies made. The reaction has been consistent – laughter. Because of their geometry, molecules are very difficult to mold, so difficult that no prototype shop has even wanted to attempt it. This is because the molding process just described works well if the part volume is *convex*. A volume is convex if a line between any two points in the interior of the volume does not leave the volume. If a solid part volume is convex, then the two halves of the mold can be removed without hanging up on any portion of the part. Note that the convexity of a part does not depend on orientation. A part that is convex remains convex regardless of how it is oriented in the mold box.

The problem with molding the molecules is that their volumes are highly *non-convex*. There are significant undulations and crevices that keep the volume from being convex. This makes it difficult to get the halves of the mold apart.

To see just how difficult the molecule-molding process would be, we tried a test with a Protein Kinase using the process shown in Figures 3-5. The resulting piece is shown below in Figure 6a.

6a

6b

Figure 6: Molded Molecule Test and showing the Transparency of the Material

The material that we used for this test was Polycon, which is a polyurethane elastomer. The material itself is fairly transparent. Figure 6b shows the same molecule held up to a bright incandescent lamp. The opacity is largely due to air bubbles trapped in the material as it hardened. This is why this process is usually conducted in a vacuum chamber.

It is difficult to tell from looking at a static photo, but this test failed to correctly capture many of the deep creases that runs through the kinase. Unfortunately, these are the areas of most interest because it is largely the depth and shape of these creases that makes the protein do what it does. In order to be an effective way to produce models, we have to do better.

The mechanical engineering design community gets around concave part problems such as this by breaking single parts into multiple parts for mass production, and then reassembling them afterwards. They do this reluctantly, because these extra steps are time-consuming and introduce another source of manufacturing error during re-assembly. For these reasons, we would like to avoid this too.

We *did* seriously consider breaking the solid volumes down into more manufacturable pieces. There are some good methods for breaking down 3D solid shapes, from cutting planes to Alpha Shapes [EDELSBRUNNER94]. Because of their complex shapes, however, these particular parts will not break down well. Because they are so highly non-convex with stringers, isthmuses, and internal cavities, there is a huge potential for small, nearly degenerate pieces that result from the splitting.

We have other reasons, though, to want to avoid building a part in several pieces:

- We wish the model to be as strong as possible. Gluing model pieces together creates an opportunity for the model to split later on. We have some experience gluing these models together, as we sometimes build complexes in pieces. Our experience is that repeated handling causes the models to start to split on the outer surface along the glue surface. These crevices gradually get larger and larger until the model either needs to be re-glued or can only be thrown away.

- We are concerned with correctly registering the pieces during re-assembly. The individual sub-assemblies must be put together correctly. Again, from our experience, this is not as easy to do as it sounds.

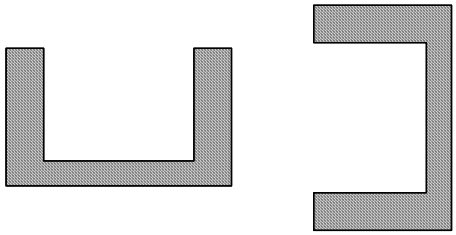
- We wish to take advantage of transparency. There are visualization advantages to being able to see into and through the model without distortion. A glue surface would hinder the transparency.

However, because mechanical design engineers also do not like to mold in multiple pieces, they often solve the difficult-to-mold problem by simply redesigning any highly concave parts to make them more convex, and thus easier to mass produce. This

is usually a good option in engineering. Oftentimes it is the general shape of a part that is crucial, not the detailed shape. Much of the detailed shape of mechanical parts is designed arbitrarily. But, we do not have that luxury. Because this is a scientific visualization application, our final part must be *data accurate*. Thus, we must figure out a way to fabricate what we have been given.

Computationally Determining Optimal Orientation

The place where the two halves of the mold meet is known as the *parting line* or *parting surface*. The first realization is that parts to be molded do not really need to be completely convex. They only need to be convex above and below the parting surface. The second realization is that they do not need to even be that convex. It is sufficient for each half to just be *vertically convex*, that is, only vertical lines would be used to test for convexity. Vertical convexity is shown below in Figure 7:



Vertically Convex **Not Vertically Convex**

Figure 7: Vertical Convexity

Neither of these parts is convex. However, the figure on the left is vertically convex, that is, any *vertical line* drawn between two points within the volume never leaves the volume. The figure on the right is not vertically convex. It is clear that vertical convexity is sufficient to make parts moldable because the two halves can be separated cleanly. It is also clear that, unlike global convexity, vertical convexity (like any other form of local convexity) is orientation-dependent.

This first research challenge then was to determine the optimal orientation of a part so that it is vertically convex, or at least as vertically convex as possible. The second research challenge was to determine the best location of the parting surface for each orientation. To do this, we applied a ray-tracing approach.

The part digital representation of the part underwent a number of trial orientations. The different orientations were spread across a *hemispherical pattern*. The hemispheric pattern used spherical coordinates, but, because this problem was vertically symmetric, it could restrict its searching to the top half-space.

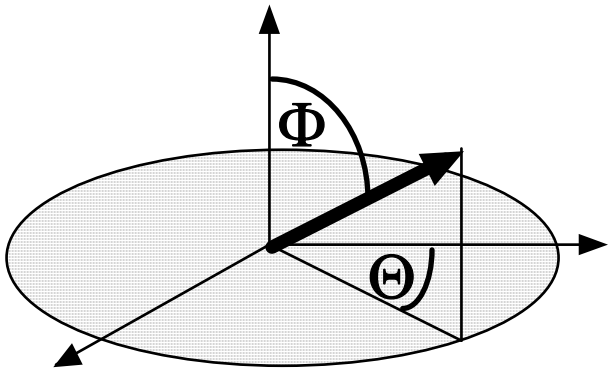


Figure 8: Hemispheric Orientation Pattern

The input to this analysis was the triangle-based STL file ([STL89]) from which the original model was fabricated. For each trial part orientation, a ray-tracing approach was used to compute a "score". Rays were fired upwards and downwards from each node in the geometric model. The first face encountered was recorded. ([FAUX79], [HOFFMAN89], [MORTENSON85])

After all rays were fired, each facet was examined by looking at the scores of its bounding vertices. The scoring was designed around a *penalty function system*, that is, the score increased, or was penalized, when the part was difficult to mold in this orientation. The Δscore of the facet was determined by three quantities:

- The Δscore was proportional to the facet's area that is blocked by other facets.
- If this facet was blocked, its Δscore was incremented by an amount proportional to the horizontal distance that this facet would have had to be translated so that it was not blocked. This is a measure of just how far off of vertically convex this facet was.
- If this facet was blocked, its Δscore was decremented by an amount proportional to the distance that the ray needed to travel to get to the occluding facet. This is to take into account the fact that a large gap between a facet and its blocking facet might not be so bad. It gives the

operator some room to bend and squeeze the mold-making material to get it out of the gap.

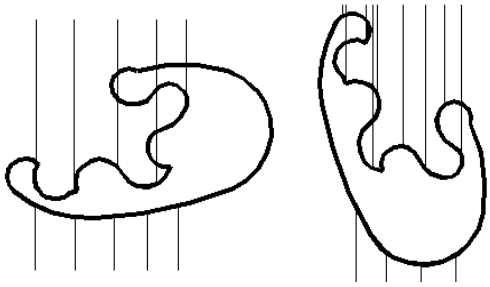


Figure 9: Vertical Rays from Each Node

Figure 10 shows some sample scores for the protein kinase in various orientations, with the scores plotted in Figure 11 as a function of Θ and Φ.

Θ	Φ	Score
0	80	10.2867
0	0	8.4243
0	60	13.2135
0	100	10.2541
0	20	12.994
0	40	15.0697
0	140	15.0198
0	120	13.1895
20	0	12.6162
20	40	16.3224
0	160	12.6548
20	20	15.7424
20	60	14.8813
20	80	11.5335
20	120	14.7173
20	100	11.5084
20	140	16.1194
40	0	14.8166
20	160	15.1354
40	60	14.3878
40	20	16.5146
40	40	16.1399
40	80	12.1119
40	120	14.3481
40	100	12.1085
60	0	13.4077
40	160	16.3418
40	140	15.9391
60	20	14.2656
60	40	13.8316
60	60	13.1055

Figure 10: Computational Results from the Ray Tracing of the Protein Kinase

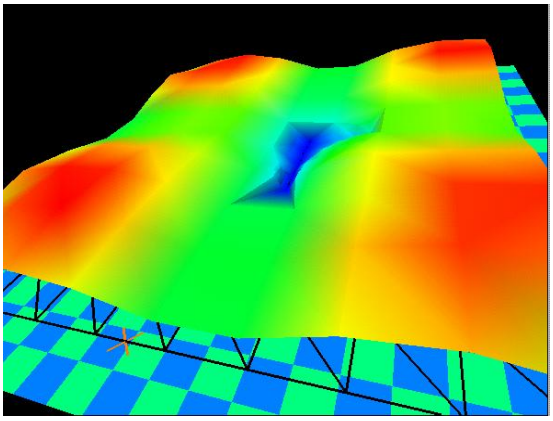


Figure 11: Plot of Ray Tracing Score (vertical and in color) versus Θ (right) and Φ (up) – blue shows the lowest scores, and thus the best orientations

A FASTER FIRST APPROXIMATION

The above process is quite time-consuming, even on a large computer. Because of this, it was desirable to find a simpler pre-process that quickly eliminated orientation possibilities that are not worth looking at.

An orthographic projection viewing of the model from above is a good first-order approximation to the needed ray-tracing. Then, a rough measure of how non-concave the part was could be had by seeing how often different portions of the model occupy the same pixels. The trick is being able to count this pixel overlap on a pixel-by-pixel basis.

Fortunately, the OpenGL graphics API has a useful construct called the *stencil buffer*. The stencil buffer allows the graphics pipeline to perform some simple pixel-by-pixel arithmetic and comparisons. Details of the OpenGL stencil buffer can be found in [SHREINER05].

We used the stencil buffer in a novel way as follows:

1. Set the pixel dimensions of the graphics window. A small size will perform each analysis quickly, but might miss some non-convex feature of the part. A large size will be more accurate, but each analysis will take longer.

2. Orient the part in its trial orientation such that the eye is looking at the top of the part.

3. Set the viewing projection to orthographic with X and Y window boundaries that exceed the size of the part. Set the near clipping plane to be between the top of the part and the eye. Set the far clipping plane to be at the parting surface.

4. Disable the Z-buffer test.

5. The stencil buffer will hold pixel-by-pixel intersection counts. Initially, clear the stencil buffer to all 0's.

6. Set the stencil function and stencil operator as follows:

```
glStencilFunc( GL_ALWAYS, 0, 1 );
glStencilOp( GL_INCR, GL_INCR, GL_INCR );
```

   This causes the count in the stencil buffer to be incremented every time this pixel is drawn to.

7. Render the part.

When the drawing is completed, the contents of the stencil buffer is uploaded and analyzed. Any pixel that has a stencil buffer value of 0 is a place where no part surface appeared. Any pixel that has a stencil buffer value of 1 is a place where only one part surface was drawn, and thus is vertically convex. A stencil buffer value of greater-than-1 will show that more than one part surface appeared under this pixel. This means that the part is not vertically convex here.

This algorithm does not tell us just *how much* the part is not vertically convex – that requires the full ray-tracing to determine. But what this technique does do is give us a *very fast rough check.* By analyzing how many pixels contain multiple regions of the part, orientations that are grossly not vertically convex can be eliminated before performing the slower ray tracing check. Also, orientations that are vertically convex or very nearly vertically convex can be ray-traced first, possibly removing the need to try lots of orientations to find one that will work.

This technique essentially uses a hardware graphics pipeline as a parallel "pixel computer" to quickly discern some information about a particular part in a particular orientation. Because this is an analysis routine only, the resulting graphics on the screen can be ignored. Or, if the user wants to be more in the loop, the stencil buffer can be used again to draw colors that highlight where a pixel contains more than one part surface.

The part can be reoriented automatically under program control, or interactively.. The object of the exercise is to maximize the amount of green that shows in the display. Thus, this technique essentially uses the hardware graphics pipeline as a "pixel computer" to quickly discern some information about a particular part in a particular orientation.

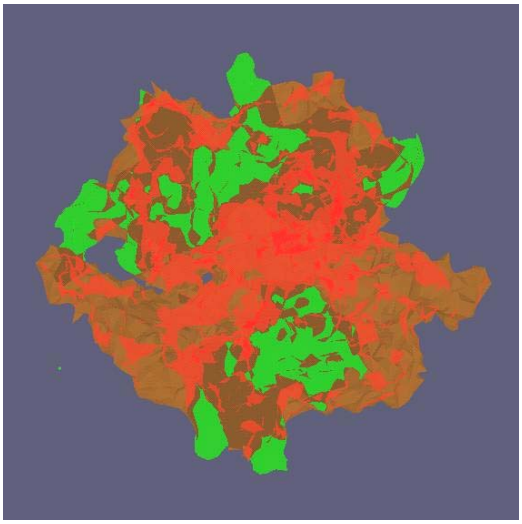


Figure 12: One orientation/parting line

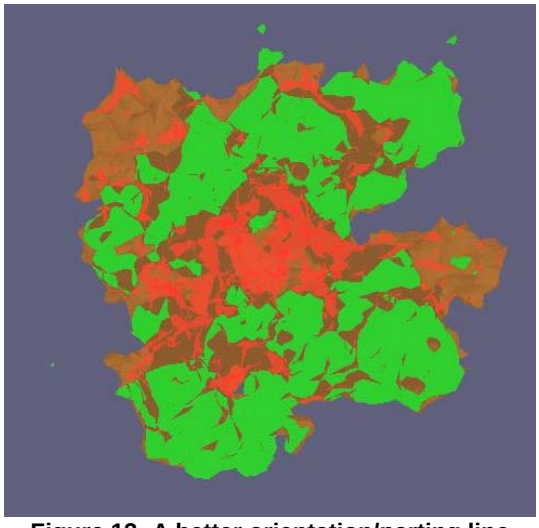


Figure 13: A better orientation/parting line

CONCLUSIONS

In applications such as fabricating data-accurate models, the model itself cannot be redesigned to meet the needs of the molding process. It must be dealt with as-is. This paper has

presented an exhaustive-search computational method for looking for optimal orientations as well as a fast, interactive, qualitative method for doing a rough check on vertical convexity.

REFERENCES

BAILEY95
Michael Bailey, "Tele-Manufacturing: Rapid Prototyping on the Internet," *IEEE Computer Graphics and Applications*, Volume 15, Number 6, November 1995, pp 20-26.

BAILEY96
Michael Bailey, "The Use of Solid Rapid Prototyping in Computer Graphics and Scientific Visualization," in SIGGRAPH '96 Course Note #37*: The Use of Touch as an I/O Device for Graphics and Visualization*, SIGGRAPH '96 Conference, New Orleans, LA, August 4-9, 1996.

BURNS93
Marshall Burns, *Automated Fabrication*, Prentice Hall, 1993.

EDELSBRUNNER94
Herbert Edelsbrunner and Ernst Mücke, "Three-Dimensional Alpha Shapes," *ACM Transactions on Graphics*, Volume 13, Number 1, January 1994, pp43-72.

FARAG1989
Mahmoud Farag, *Selection of Materials and Manufacturing Processes for Engineering Design*, Prentice-Hall, 1989.

FAUX79
I.D. Faux and M.J. Pratt, *Computational Geometry for Design and Manufacture*, Ellis Horwood Publishers, 1979.

HALLUM97
David Hallum, "Fundamentals of Injection Molding," *Manufacturing Engineering*, June 1997, pp68-72.

HART96
David Hart, "Putting Proteins into Researcher's Hands," SDSC *Gather / Scatter*, Vol 12, No 1, January 1996, pp12-13.

HOFFMAN89
Christoph M. Hoffman, *Geometric & Solid Modeling*, Morgan Kaufmann Publishers, Inc, 1989.

KRYGIER94
J. B. Krygier, "Sound and Geographic Visualization," *Visualization in Modern Cartography*, Pergamon Press, pp149-166.

MENGES93
Georg Menges and Paul Mohren, *How to Make Injection Molds*, Hanser Publishers, 1993.

MORTENSON85
Michael E. Mortenson, *Geometric Modeling*, John Wiley & Sons, 1985.

PTC96
Pro/MOLDESIGN User's Guide, Parametric Technology Corporation, 1996.

PTE96
Silicone Rtubber Molding Process, PTE Distribution, 1996.

SALISBURY97
J. Kenneth Salisbury and Mandayam Srinivasan, "Phantom-Based Haptic Interaction with Virtual Objects," *IEEE Computer Graphics and Applications*, Volume 17, Number 5, September 1997, pp6-10.

SCHULTEN97
http://www.ks.uiuc.edu/Research/bio_ener/LH_2

SHREINER05
Dave Shreiner, Mason Woo, Jackie Neider, and Tom Davis, OpenGL Programming Guide, Fifth Edition, Addison-Wesley, 2005.

STL89
"Stereolithography Interface Specification," 3D Systems, Inc, October 1989.

WARNER96
Thin Wall Reaction Injection Molding: A Low Pressure Process for Rapid Prototyping, Warner Technologies white paper, 1996.

WHELAN90
Tony Whelan and John Goff, *Injection Molding of Thermoplastic Materials, Volume 2*, Van Nostrand Reinhold, 1990.