A new approach to mold making

Fabian Ruiz

Abstract—In rapid prototyping, one aspect that is important, is mold design. Actual approaches to make a mold in a computational way, works well, but usually if the object is parametrized otherwise in some case does not have the desire precision. Beside this, using the actual approach, tends to be complex, because some steps are difficult to fix in case there are problems. The focus of our work is to create an alternative methodology that works with every type of object, that is at least equally fast, and precise, and avoid these steps. The result we got, were optimistic, and it could impact the area of rapid prototyping to make molds in an easier way. Finally, our future work, is to make our approach automatized.

MOTIVATION

In rapid prototyping [1], one aspect that is important, is the creation of mold [2], instead of the 3D printing[3] of the object itself, in order to replicate the object, in a simpler and faster way, besides the possibility to use other type of materials. Even though there is a variety of studies that focus on mold design in a efficient and structural way [4], there is few information on how to create mold in a computational way.

On the other hand, there are algorithms used in the mold making process, but tend to be slow, and in some cases, the precision is not the desirable one. Nevertheless, this algorithm, can be really precise and fast, but only if the object is parametrized, which mean, that the object has to be represented by geometry and interrelations.

Besides this, some steps of the traditional approach to use these types of algorithms with non-parametrized objects, do not works always.

Therefore, the focus of this investigation, is to design a new approach, to create molds in a computational way, that is precise, fast and avoid these steps, to be able to be used on the rapid prototyping area in an easier way.

- *Fabian Ruiz is with Austral University of Chile*
-

RELATED WORKS

Varies of algorithms exist to create mold, one of the most studied, are the Boolean operation, which are a type of operation to make intersection, difference, and intersection, beside other type. However, there are two limitations, the mesh [5] must be triangularized [6] and there cannot be coplanarity between points, which cause precision loss [7]

In the case of mold making of parameterized objects, it tends to have a great precision, because the mold can be parametrized,in a way that is focused on precision, but, one of the most important limitations of this, is, there are a variety of objects that, when are parametrized, have a precision loss [8].

Because of this type of limitations with the Boolean operations and the parametrization of objects, it was decided to create a new approach focused on precision and speed, that can be used on any type object, even if it is not parameterized.

METHODOLOGY

The usual approach to make mold, is to reduce the count of polygons, correct normal, triangularized mesh, use the difference Boolean operation between the object and a cube, and cut in two halves (Fig 1). These five steps are subdivided in three sections, preprocessing, processing, and post processing As mentioned in the Motivation section, some steps of this methodology do not always work, one of these steps is, correct normal, because, when the count of polygons are reduced, sometimes complex manifold are formed on the object, which cannot be repaired or are difficult to repair.

Other step that cause troubles, are the mesh triangularization because not every object can be triangularized, and the Boolean operation because is slow if is not used in an optimal way, that is why it was decided to create an approach, that avoid these 3 steps, to create the mold.

To do this, the approach created, had five steps (Fig 1), flip normal, create a cube for molding, combine the cube and the inverted object, intersect a plane and close the surface, dividing these five steps, in two sections, pre-processing, and processing

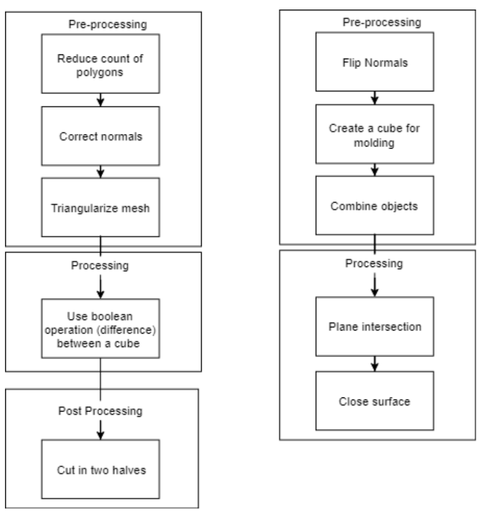

Figure 1. Left traditional approach pipeline, Right, new approach pipeline.

Step 1: Flip normal. To create a mold, that avoid the use of Boolean operation, it was decided to flip the normal of the object, to get the inverse of the object

Step 2: Create a cube for molding. After the inverse of the object is obtained, the next step, is to create a cube that is bigger

than the object, to place the inverse of object inside the cube.

Step 3: Combine Object. Once the inverse of the object is inside the cube, they have to be combined and become one object, in order to be processed

Step 4: Plane Intersection. To get the same effect that is obtained in the traditional approach of making mold, the plane intersection has to be in the middle of the inverse object

Step 5: Close surface. When the object is cut with the plane, the surface is completely open, so it has to be closed to create a functionally mold.

This new approach was tested step by step, in Meshmixer, a software created by AutoDesk. Meshmixer was chosen, because is a robust and well-maintained software, that has all the operations needed for the new approach to work well. Then it was compared with the traditional approach. To compare the two approaches, 5 3D objects were downloaded from thingyverse, a page that has a variety of 3D objects with free royalty and free of use for various purposes, and one extra object, created by ourselves, that was parameterized, with a final count of 6 objects, which can be seen in the table 1, ordered by triangle count. For every object, two mold were made using Meshmixer, one for each approach, and then the speed and precision was compared.

Object	Triangle Count
Warrior Figure	58542
Hand	100523
Shoji Piece (Parameterized)	120556
Toy train	155312
Simple Engine	374885

Table 1
Table to test captions and labels

From the table 1, is important to note, that, making a mold of a complex engine is useless, because it would not work for 3D printing, but, it was to stress the different approaches, and test the worst-case scenario with the complex object, and test the average scenario with the first five objects.

To measure the speed, it was decided to focus only on the time using for processing, because the time used for preprocessing, and post processing is negligible. Beside this, another software had to be used, to take the time in a precise way, because Meshmixer does not have a feature to measure the time of processing. Even though there were time difference between the process start and the start of the timer, it was a negligible amount of time, in the order of 0.01 seconds. To make the a more robust comparison, every test with each approach, was made 10 times, and then the average of the time to process was calculated for each test. To see if the new approach was better in terms of speed, a speedup formula was used between the traditional approach and the new approach:

$S_p = T_{traditional}/T_{new}$

Then, a graph was made, using the speedup, and the count of triangles, to see if the new approach can escalate with a large count of triangles.

On the other hand, to measure the precision difference, it was decided to visual compare the two molds for each object, because, even though, one could take in count the difference in the quantity of triangle and polygons of the final mold, is not necessarily more precise. Beside this, only the pieces that were visibly different, were shown in the results.

Results

The result obtained were a graph, that is shown in the figure 2, and the comparison of the mold objects, from figure 3 to 6.

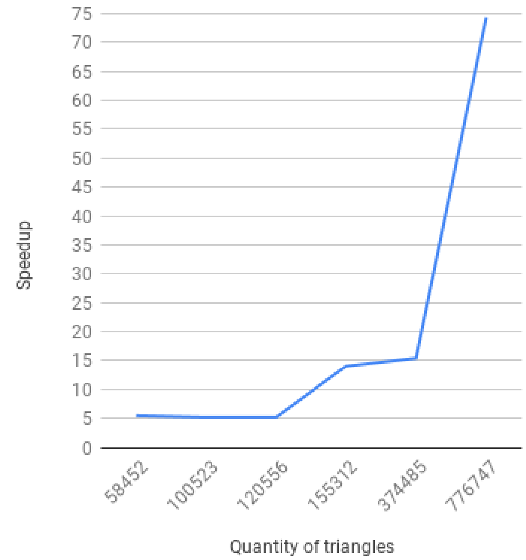

Figure 2. Graph Speedup v/s triangles

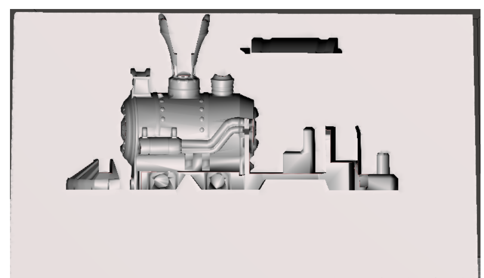

Figure 3. New approach, toy train

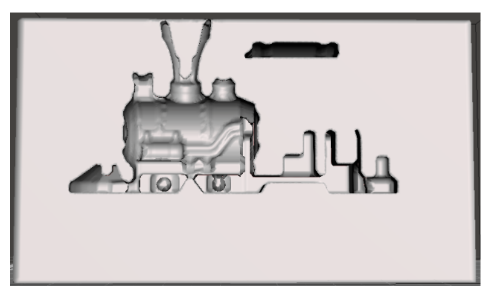

Figure 4. Traditional approach, toy train

Discussion

We can see in the graph (Figure 2), that this new approach is between 5 and 15 faster than the traditional approach in the average case, and almost 75 faster in the worst case scenario, this is probably because, the more triangle an object has, in the case of the traditional approach

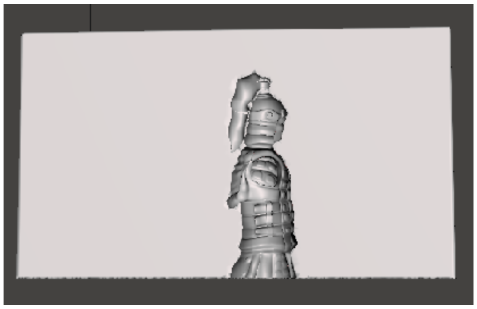

Figure 5. New approach, warrior figure

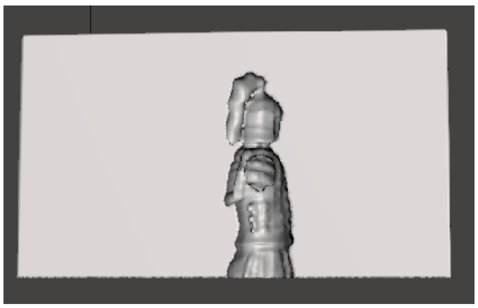

Figure 6. Traditional approach, warrior figure

it has to process more intersections between voxels, while the new approach scale in a linear way, because it use only a plane instead of a cube to do the intersection. On the other hand, the precision on almost every object, did not vary like we expected to, but in some cases, in which the object has little marked details, the variations between the traditional and new approach was visibly enough. While the traditional approach appears to be blurred (Figure 3 and 5) on the details, the new approach, has details marked in a visible way (Figure 4 and 6).

We conclude that, our approach, can be used in the rapid prototype area, because is faster in any case, and is more precise in some cases, than the traditional approach. Beside the performance, the impact on the area could be important, since it use alternative steps to solve cumbersome and complex problems, which mean it could be easier to make mold for people that is starting to learn about this topic.

One of the biggest limitation of our work, is that the approach we made, was only tested in Meshmixer, and the software has a close surface algorithm already implemented, which not every 3D editor has. Beside this, the scope of our work, is limited to mold design only, since using this approach for other type of design, would not work correctly, because it was specifically made to create mold.

Finally, for the future works, the idea is to automatize the process with graphical process libraries like CGAL [9] or VTK [10], to be able to simplify the usage, and make it so it can be used in the rapid prototype area, with less experienced user in software editors.

BIBLIOGRAPHY

[1] F. Liou (2008). Rapid prototyping and engineering applications. Boca Raton: CRC Press.

[2] S. K. Nayak. (2012). Fundamentals of Plastic Mould Design. McGraw-Hill

[3] C. B. (2013). 3D Printing: The Next Industrial Revolution

[4] D. O. Kazmer. (2015). Injection Mold Design Engineering (Second Edition). Carl Hanser Verlag.

[5] P. Shirley,M. Ashikhmin,S. Marschner. (2009). Fundamentals of computer graphics. (Third Edition) A K Peters/CRC Press;

[6] Computer Graphics and Geometric Modelling. (2005). Springer Science. & Business Media

[7] C. Quammen , C.Weigle , R.M., II, Taylor,. (2011). Boolean operations on surfaces in VTK without external libraries. Insight J. 1-12.

[8] Yasser Zarei (2012) The Challenges of Parametric Design in Architecture Today: Mapping the Design Practice

[9] Board, C. (1995). ”CGAL, Computational Geometry Algorithms Library” Retrieved from https://www.cgal.org/

[10] Schroeder, Will; Martin, Ken; Lorensen, Bill (2006), The Visualization Toolkit (4th ed.), Kitware