Math 4020
3D Printing Group Final Report

Gayan Abeynanda Brandon Oubre Jeremy Tillay Dustin Wright

Wednesday 29th April, 2015

Introduction

3D printers are a growing technology, but there is currently no objective and widely-used method of comparing the quality of 3D printers.

The goal of our project is to develop a good metric for measuring the quality of these printers. The essential task is to compare an STL file's ideal representation in the 3D world, which is the file sent to the 3D printer to print, to a real-world printed object. If the objects are very similar, the printer is essentially very good. If there are significant differences due to low resolution or because parts of the object fall off, then it is a very bad printer.

Our task mathematically is to compare an STL file to a real-world scan of these objects. We have to ensure that our comparison is "fair" in the sense that it compares them once they are aligned and scaled appropriately. If someone scans the picture but misplaces or misrotates it slightly, but the object is a great representation of the ideal object, it should not see the skew the metric. Similarly, a bad printed object shouldn't be placed in such a way that the metric does not identity how truly different the idealized and ideal objects are.

Then, once the objects are aligned and scaled appropriately, we can define a metric to compare them.

Approach

Our approach has two inputs: An STL file representing an object to be 3D printed and the 3D scan of the actual printed object. The scan will be provided to our algorithm as either a series of 2D cross sections of the actual object or as a 3D intensity matrix. We will refer to this set of cross sections as the actual image series and to the 3D intensity matrix as the actual intensity matrix. From the STL file we will generate another 3D intensity matrix that we will refer to as the ideal intensity matrix. In order to align the actual matrix (or series) to the ideal matrix, we will use an algorithm we developed that is described in a later section. Once the two objects have been aligned, we can then apply a metric to them in order to determine how "close" the objects are and assess the printer's quality.

Ideal Matrix Creation

One method that we have proposed for creating an ideal intensity matrix is as follows: First, we want to assume that all values of both matrices are on a predefined interval. To achieve this, we can simply apply the following transformation to the actual 3D intensity matrix where $A_{i,j,k}$ is the i, j, k^{th} coordinate of the actual intensity matrix and A_{max} is the largest value in the matrix:

$$A_{i,j,k} \leftarrow \frac{A_{i,j,k}}{A_{max}}$$

This transformation guarantees that all points in the actual intensity matrix are now scaled down to the $[0, 1]$ interval. Also, the intensity matrix is a negative image. Intensity values are lower where the object existed in the scan. For intuitive purposes and for later calculations, we can transform this into a positive image where higher values represent object presence by applying the following transformation after the previous transformation:

$$A_{i,j,k} \leftarrow 1 - A_{i,j,k}$$

Now that we have applied some transformations to the actual intensity matrix, we can proceed with creating an ideal intensity matrix. We propose accomplishing this by creating a 3D matrix that is a "pixelated" version of the STL file. That is, we create a matrix of the same dimension as the actual intensity matrix, and if given the scale information from the actual matrix we assume the same scale for the ideal matrix. Each cell in the ideal matrix now represents a point in 3-space. One way to fill this matrix is for each cell in the matrix, we test if the associated point is on the interior or exterior of the STL file object. If on the interior, we assign the point as 1, if on the exterior we assign the point as 0.

Matrix Alignment

Our method of alignment is based off of the assumption that we are printing objects that can be laid flat. We are also assuming that when the object is scanned, the person placing the object on the scanner has placed it on the right side and with a close angular orientation to the orientation of our ideal file.

Assuming this, we generate a matrix of "ideal" values, where the matrix $A_{i,j,k} = 1$ if there is supposed to be matter printed at that point and 0 otherwise. The size of this matrix corresponds to the amount of pixels in our scanner.

Then we have our 3D matrix that comes from the scan of the actual object, which will will denote $B_{i,j,k}$.

Now, we are going to create functions defined over $\mathbb{R}^3$ so that we can perform continuous operations such as scaling and rotating in order to properly align the objects.

Let $f(x, y, z) = A_{round(x),round(y),round(z)}$. So if the matrix A is essentially a cloud of points evaluating whether there is mass or not at a discrete set of points, our function is a filling in of the 3-space with values 1 or 0 corresponding to the values at these integer points.

Let $g(x, y, z) = 1$ if $B_{round(x),round(y),round(z)}$ is above a certain numerical threshold σ that we determine and 0 otherwise. This numerical threshold exists because B is a 3D matrix corresponding to a scan of a physical object. We must set the threshold to correspond to a number that would realistically come from a scan of air. Any number above that would presumably correspond to a point with matter.

Now recall we can find the center of mass (x', y', z') of a function $f(x, y, z)$ by calculating:

$$(x')_f = \frac{\iiint f(x,y,z)x dV}{\iiint f(x,y,z) dV}$$
$$(y')_f = \frac{\iiint f(x,y,z)y dV}{\iiint f(x,y,z) dV}$$
$$(z')_f = \frac{\iiint f(x,y,z)z dV}{\iiint f(x,y,z) dV}$$

We will first shift our functions by the center of the mass, so that now the center of mass of the object will be moved to the center. A shift of the object in the direction $\langle -x', -y', -z' \rangle$ is given by

$$f_s(x,y,z) = f(x + (x')_f, y + (y')_f, z + (z')_f)$$
$$g_s(x,y,z) = g(x + (x')_g, y + (y')_g, z + (z')_g)$$

Doing this moves the center of mass to the origin for both objects. We do this because our assumption is that when the objects are properly aligned, they will have a center of mass at the same point. Also, we move this point to the origin, so that when we rotate around the z-axis, the center of mass remains at the origin. This way, we avoid having to iterate between shifting, rotating, reshifting, rotating, etc.

Finally, we do a rotation of the object. A rotation of g_s by θ degrees clockwise can be expressed as $g_s(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta, z)$.

Now we must find the rotation that will maximize overlap between the objects, aligning them as closely as possible. We can do this by finding

$$\max_{\theta} \iiint [f_s(x,y,z) * g_s(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta, z)]dV$$

This maximum will occur whenever there is a maximum overlap between the two functions, and corresponds to the θ that we must rotate g_s by to properly align the functions. We can search within a small range around 0 since our assumption is a human carefully placed the object and got somewhat close to maximal alignment. This is computationally very slow, but can get closer to perfect alignment of an ideal and real object with enough time. There is potential for improvement, perhaps a more iterative process we found in our research that we did not have time to implement.

Alternative approach to alignment - Alignment via cross correlation of Radon transforms

In this approach, the alignment of the actual intensity matrix (corresponding to the printed object) to the ideal intensity matrix (reference, corresponding to the actual object) is achieved with the use of a method developed for 3D rotational alignment of particles in microscopy [1, 2]. The method is based on cross-correlations of two and three dimensional Radon transforms. The technique can essentially be described as follows: All possible two dimensional projections of the actual object are calculated and used as references for alignment of the experimental projections of the printed object by cross-correlation techniques. The Radon transforms provides the necessary mathematical tool for the alignment since the three-dimensional Radon transform contains the same information as a complete set of two-dimensional projections, in a much smaller data set and facilitates simultaneous implementation of translational and rotational alignment.

The translation and the orientation of the printed object relative to the reference (actual object) is determined simultaneously by calculating a 5 dimensional (3 Euler angles α, β, γ corresponding to the orientation and the magnitude and the argument of the shift vector, $r$and η) cross-correlation function. The values of α, β, γ, η and $r$for which the cross-correlation function assumes its maximum define the alignment of the printed object relative to the reference.

Definition:

Given a 3 dimensional function $f(r)$, $r = (x, y, z)$, its Radon transform is defined as

$$R(f(r)) \quad \hat{f}(p,\xi) = \int (r)\,\delta(p - \xi.r)\,dr \quad ;$$

$$\delta(p - \xi.r)$$

determines the plane over which the integration is carried out

$$\xi = \begin{pmatrix} cos(\theta)\sin(\phi) \\ cos(\phi) \\ sin(\theta)\sin(\phi) \end{pmatrix}$$

is the unit vector orthogonal to the plane. Analogously, given a 2 dimensional function $g(r)$, $r = (x, y)$, its Radon transform is defined as

$$R(g(r)) \quad \hat{g}(p,\zeta) = \int (r)\,\delta(p - \zeta.r)\,dr \quad ;$$

$$\delta(p - \xi.r)$$

determines the line over which the integration is carried out

$$\xi = \begin{pmatrix} sin(\epsilon) \\ cos(\epsilon) \end{pmatrix}$$

is the unit vector orthogonal to the line

The two-dimensional Radon transform can be calculated by calculating all one-dimensional projections (in small angular increments) of a two-dimensional image. A

three-dimensional Radon transform can be calculated in two steps, by first calculating all two-dimensional projections perpendicular to the $x-z$plane, followed by a two dimensional Radon transform of all of the two-dimensional projections.
Implementation:

Given the three dimensional Radon transform of the reference (actual object), represented as $\hat{f}(p,\phi,\theta)$ which is identified as a set of integrals over lines defined by the angles $\theta$of the two-dimensional projection and a second angle ϕ defined by the projecting direction within the plane at angle $\theta$and the two-dimensional Radon transform of the projections $\hat{g}(p,\epsilon)$ identified as the set of integrals over lines at angles $\epsilon$the method is implemented through a series of computations.

First, the two-dimensional Radon transform of a projection at Euler angles $(\alpha,\ \beta,\ \gamma)$ is extracted from the three-dimensional Radon transform $\hat{f}(p,\phi,\theta)$. Let $\hat{r}_{\beta,\gamma}\left(p,\ \alpha'\right)$ be this two-dimensional Radon transform extracted from $\hat{f}(p,\phi,\theta)$ and let $\hat{g}(p,\epsilon)$ be the two-dimensional Radon transform of one projection, where $\alpha' = \alpha+\epsilon$. Then angles ϕ,θin the three-dimensional Radon transform belong to the line at angle ϵin the two-dimensional Radon transform of the projection at angles $\alpha,\ \beta,\ \gamma$.

Then to find the orientation and translational relationship of the printed object relative to the reference, the cross-correlation function of the Radon transforms is defined. To achieve this, the shifting property of the Radon transform has to be used:

$$R\left(f(r-a)\right) \quad \hat{f}(p,\xi) = \int (r-a)\,\delta\left(p-\xi.r\right)dr = \int (s)\,\delta\left((p-\xi.a)-\xi.s\right)ds$$

The cross- correlation function depending on the three Euler angles $\alpha,\ \beta,\ \gamma$and a shift vector of magnitude r and argument $\eta$is then defined by;

$$c\left(\alpha,\ \beta,\ \gamma, r, \eta\right) = \iint \hat{r}_{\beta,\gamma}\left(p,\ \epsilon\right)\ \hat{g}\left(p-rsin(\epsilon+\eta),\epsilon+\alpha\right)\ dp\ d\epsilon$$

For higher computational efficiency this cross-correlations are calculated by a multiplication of the Fourier transform of the corresponding Radon transforms followed by inverse radial Fourier transformation.

The angles $\alpha_0,\ \beta_0,\ \gamma_0,\ \eta_0$ and the value r_0 for which the cross-correlation function assumes its maximum provides the alignment of the printed object relative to the reference.

Metric

Once the objects are aligned, there are a fair number of ways to define a metric evaluating the printer. First, we can print a set of objects that adequately test the printer's capabilities. Such test objects might include something with a long extraneous edge that might easily break off, a very bumpy object, or a simple sphere/cube to test its capabilities at a very basic level.

Then, we can take $\int |i(x)-r(x)|dx$ where $i(x)$ is a function of the ideal object that has been aligned and scaled. It will be 0 where there is empty air and 1 where there is supposed to be matter. $r(x)$ is a function of the scan of the printed structure that has been aligned and scaled. It can again be 0 where there is empty air and 1 where is supposed to be matter.

Alternatively, instead of 1, we can assign different values that will highlight the importance of certain parts of the structure. For example, the center of a sphere or cube could have a higher value if we care more about the 3-d printer ensuring that the core of the object is printed. If the center is hollow, the integral will be much larger. If we print an objects with bumps, the bumps could have relatively higher values if we really want to highlight whether the printer adequately

Conclusion

Our group has developed a methodology that will assist in evaluating the quality of 3D printers. Given an object's STL file and data from the 3D scan of a printed object, we can align the scanned data to match the data from the STL file and then compare the actual object to its idealized version. This comparison can then be used to assess the quality of the print and used in determining the overall quality of the printer.

References

[1] Radermacher, M. (1997). "Radon Transform Techniques for Alignment and Three-Dimensional Reconstruction from Random Projections". Scanning Microscopy Vol 11, Pages 171-177.

[2] Radermacher, M. (1994). "Three-dimensional reconstruction from random projections: orientational alignment via Radon transforms". Ultramicroscopy Vol 53, Pages 121-136.