university of groningen / faculty of science and engineering

Anamorphic Sculptures with Shading

Master's Research Internship

University of Groningen

July 9, 2024

Author:
Tom Couperus

Primary supervisor:
Prof. Dr. Jiří Kosinka

Secondary supervisor:
Dr. Steffen D. Frey

Abstract

Anamorphic art uses perspective and mirrors to view a distorted image or sculpture in the correct way. Normally, making anamorphic art is a difficult task where errors can easily occur, since the artist has to create the image while thinking of their desired perspective. So, for centuries anamorphic art has been relegated to a curiosity.

In recent years interest in the topic has been renewed and various papers have been published on anamorphic art. One such paper [L. Pratt, A. Johnston, and N. Pietroni, 2023] has advanced the creation of anamorphic sculptures by computer simulation to work with free-form 3D shapes as well as reflective and refractive materials. However, the paper does not consider light sources or normals, which can lead to odd shadows or lighter areas being introduced to models where it is not desired. Our project expands on the original project by developing a simplified version with limited reflection depth and adding lighting. We experiment with multiple approaches to solve this problem. We conclude that there are exact solutions for some specific cases, but that an approximation by minimizing the angle between deformed normals and what they should be is the best, most general solution. However, due to the approximation, there remain issues with lighting to be resolved.

Contents

1 Introduction

Anamorphic art uses perspective and mirrors to view a distorted image or sculpture in the correct way. It has a rich history since Leonardo da Vinci drew the first anamorphic image in his Codex Atlanticus [1]. A well-known example of anamorphosis in modern times are driving instructions painted on the road. These instructions look elongated when viewed from above or from the sidewalk, but when looked at from a car, they look completely normal. This is due to the driver's lower perspective when approaching the sign painted on the floor. Of course, more complicated images, such as the skull in the famous painting "The Ambassadors" by Holbein (Figure 1), require the artist to create the image while thinking about their desired perspective, instead of simply elongating a letter. This is a difficult task where errors can easily occur, so for centuries anamorphic art has been relegated to a curiosity.

Figure 1: "The Ambassadors" by Hans Holbein the Younger. In the left image, the painting looks normal, except for the odd diagonal streak near the bottom. When looking at this image from the correct viewpoint, the skull reveals itself.[1]

In recent years interest in the topic has been renewed and various papers have been published on anamorphic art. Some of these papers define anamorphosis mathematically [2], while others have looked into computing the distorted versions of objects [3]. The first of these approaches to use 3D shapes and mirrors to calculate the deformed version of objects was proposed in [4], but it was limited to spheres and other geometric primitives.

A paper from last year [5] has advanced the computation of anamorphic sculptures to work with free-form 3D shapes as well as reflective and refractive materials. In this paper the authors create the anamorphic representation of the desired object by a ray-tracing method [6]. Their system uses the observer's point of view, the various deformation surfaces and the 'correct' sculpture mesh to generate the mesh of the deformed sculpture by tracing rays from

[1] Images from `https://en.wikipedia.org/wiki/Anamorphosis`, accessed June 2024.

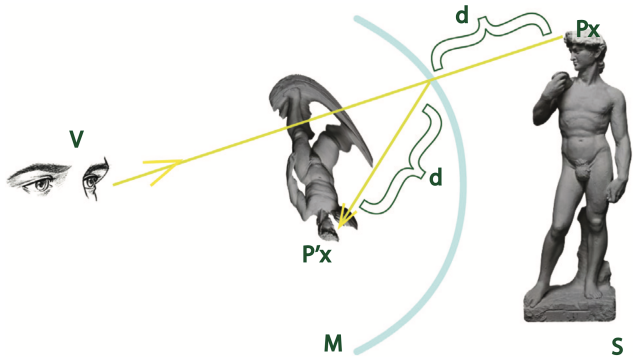


Figure 2: The sculpture is deformed such that each point stays in the target position expressed by the reflected ray and the distance d. Image and caption from [5].

the view point to the 'correct' object. When a ray goes through a reflective or refractive surface, the reflected or refracted ray is calculated, and the deformed vertex can be placed along any point on the ray. To preserve the normals reasonably well, the authors calculate the distance d between the 'correct' object and the first intersection with a reflective or refractive surface. This distance d is then used to determine the final position of the deformed vertex along the last reflected or refracted ray, as shown in Figure 2.

However, the paper does not consider light sources or normals, which can lead to odd shadows or lighter areas being introduced to models where it is not desired. An example of these distorted normals introducing lighter areas where there were none is shown in Figure 3. In this top-down perspective view, a flat plane is deformed through a curved mirror, creating a deformed version of itself. A directional light is placed at infinity, shining parallel to the flat plane at a slight downwards angle. This way, the light shines in the same direction onto the deformed version. As Figure 3 shows, a lighter area is introduced in the deformed version of the plane where it should not be, according to the 'correct' model. This is due to the curvature of the deformed object, which is now able to reflect some of the light.

To determine if it is possible to remedy this, our project expands on the original project by developing a simplified version with limited reflection depth and adding lighting. Through experimentation with the deformation algorithm, we attempt to determine a way to better preserve the relative normals of the original object.

In Section 2 we describe the various approaches we tried, and why some of them were rejected. The results of our final method are discussed in Section 3. Section 4 concludes our research, and Section 5 proposes further avenues of research.

2 Methodology

Our testing environment is a simplified version of the setup found in [5]. Figure 3 shows our testing scene from a top-down perspective view, but we replace the highlighted plane by other objects, such as a cube or a sphere. The scene consists of a viewpoint and an object as it should be seen in a reflective surface or through a refractive surface. This surface is placed between the viewpoint and the object. For our testing purposes, we limited ourselves to a single reflective mirror, that is convexly curved around the y-axis with a radius of curvature

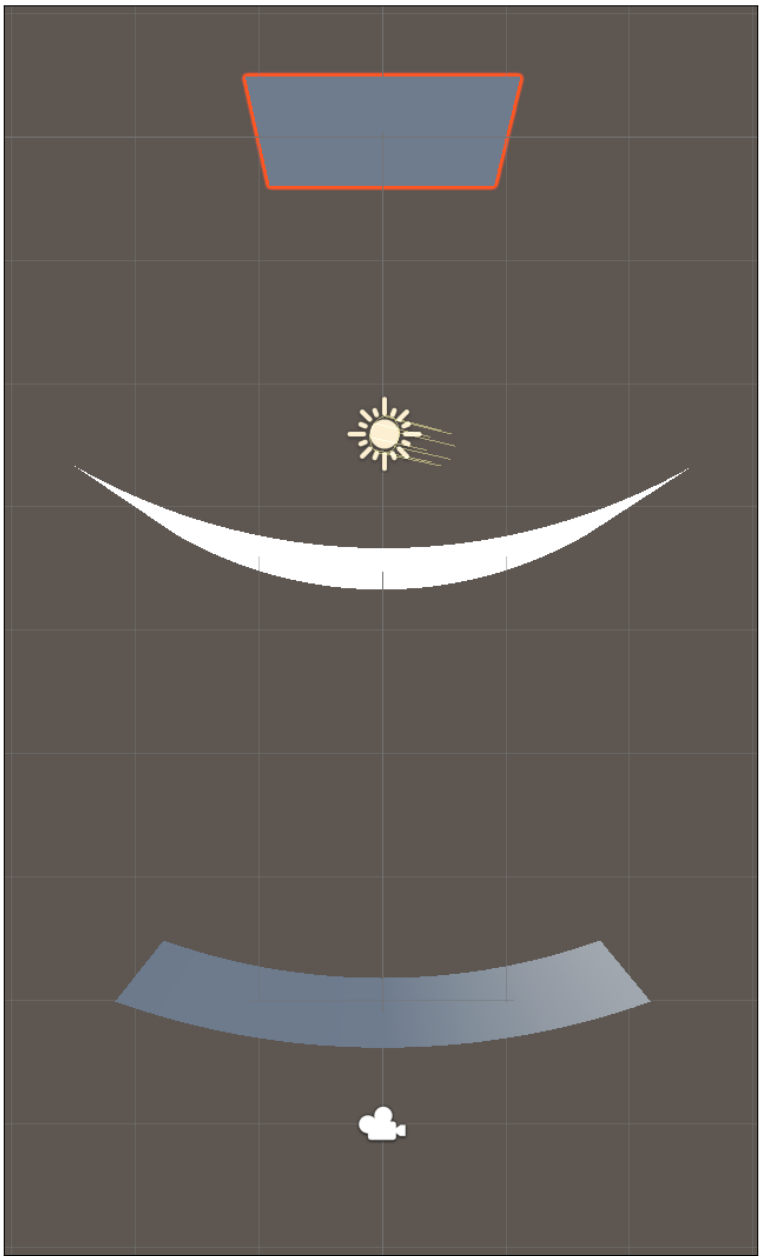


Figure 3: A ‘correct’ plane (highlighted), a curved mirror (white), a directional light at infinity shining parallel to the plane (sun icon), and a viewpoint (camera icon). The plane is deformed through the mirror, with the directional light shining parallel to it, not reflecting any light. The deformed version on the bottom is lighter on the right.

of 2. We have a directional light placed at infinity, shining parallel to the mirror's plane, if the mirror would be completely flat. This ensures that if a flat plane were placed behind the mirror, facing the viewpoint, its diffuse component would be 0, as the light's direction is orthogonal to the plane's normal. If the deformed version matches the normals of the original, it also would have no diffuse lighting. However, if the deformed normals do not match the 'correct' normals, we will find lighter areas where there should be none.

For the lighting, we only consider the diffuse component, as defined in the Blinn-Phong shading model. This is calculated for each pixel as $I_d = k_d(\overline{L} \cdot \overline{N})i_d$. In this equation, $\overline{L}$ is the light's direction, $\overline{N}$ is the normal's direction, k_d is the object's diffuse reflection constant, and i_d is the light's diffuse intensity. Both $\overline{L}$ and $\overline{N}$ are unit vectors. Placing a directional light at infinity allows us to only optimize for the normal direction, since $\overline{L}$ is now a constant as well.

We analyse the results of our methods by mapping the angle between the optimized normals and the 'correct' normals to a grayscale colour. An angle of 0° is mapped to a dark colour, and an angle of 180° is mapped to a lighter colour. Figure 4 shows the colour that each angle is mapped to on a circle. The shader does not discern between clockwise and counter-clockwise rotations.

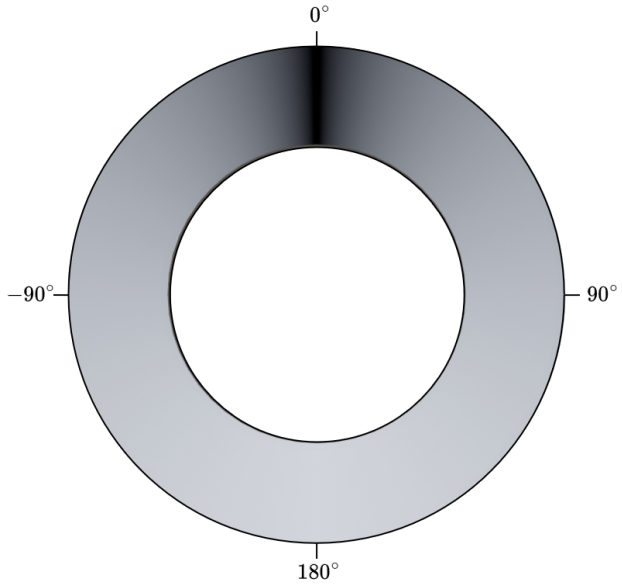


Figure 4: Annulus of colour mapping of angular deviation of anamorphed normals relative to the normals of the 'correct' model.

We create the deformed version of the object placed behind the reflective surface using the algorithm developed in [5] and described in Section 1.

As we use the Unity game engine and the C# language for our project, all objects in this project are represented by triangulated meshes. The Unity engine provides a simple framework to work with these meshes, as well as an easy-to-use scene editor and a customizable editor UI.

Our goal is to determine whether it is possible to deform an object in this way and fully restore its relative normals such that the lighting is not changed by the deformation through

the mirror, as is the case with the current approach. To do this, we first look for a single-pass solution. However, as we will show in the sections below, we were unable to find such a solution to this problem that fits more than very specific cases. To still arrive at better fitting normals, we also experiment with an iterative approximation approach.

2.1 Two-dimensional Approach

As our setup only contains a mirror surface that is curved around the y-axis, we first tried restoring the relative normals in the plane orthogonal to this axis. The mirror is curved around the y-axis, so our plane of curvature is the plane formed by the x and z-axes.

In Figure 5, the points P_n are the vertices of the 'correct' object, and the points P'_n are the corresponding vertices distorted through the mirror. The P''_n points indicate the ideal deformed positions of each vertex, such that the normals are restored. A ray I_n is traced from the viewpoint V to each vertex P_n. The reflection points M_n and reflections R_n of these rays are also shown.

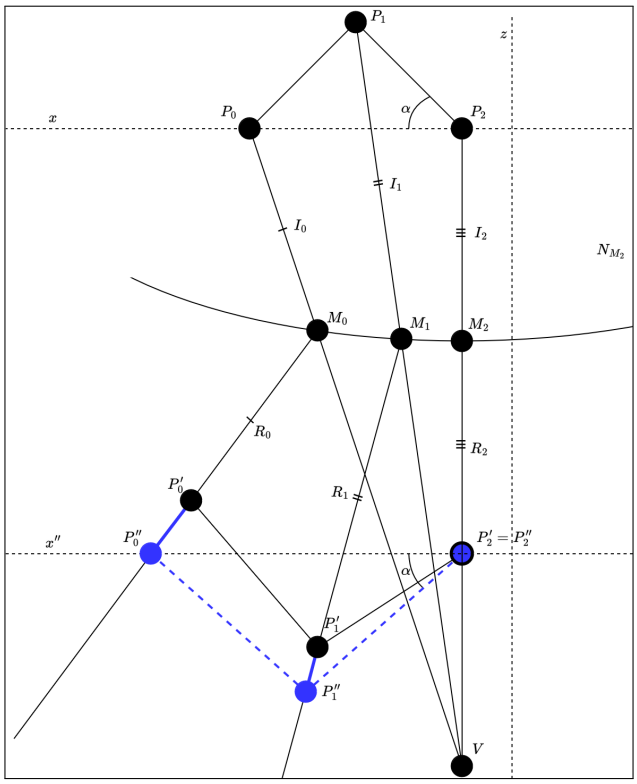
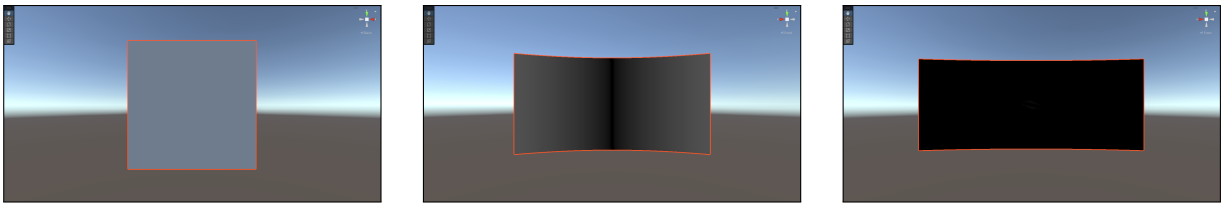


Figure 5: Diagram of 2D optimization.

One vertex P_c is chosen as the reference point for all further calculations. Next, we calculate for every point P_n ($n \neq c$) the angle α between the x-axis through P_c and the line between the two points. We then make a line from P'_c that is α degrees rotated away from the x'-axis through P'_c, mirroring the rotation in the 'correct' model. This line is then intersected with the reflection ray R_n of the vertex to obtain the ideal deformed position P''_n. This intersection is calculated in the xz-plane and thus needs to be projected up into

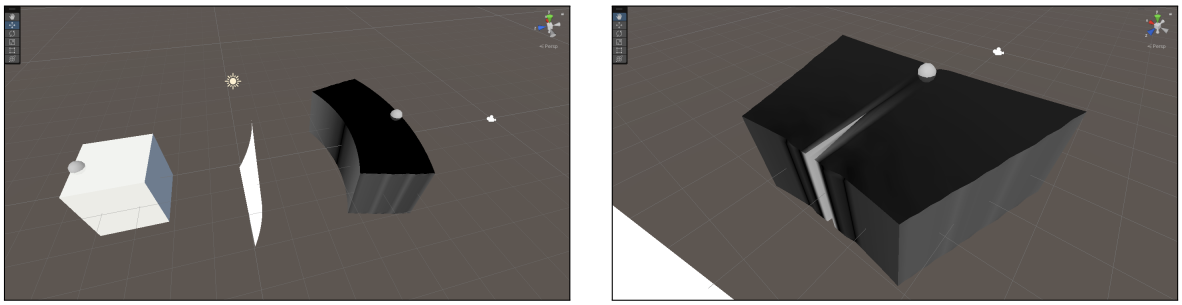
3D. We do this by scaling the normalized 3D reflection ray by the factor $\gamma = \frac{P''_{n,x} - M_{n,x}}{P'_{n,x} - M_{n,x}}$, which uses the x-components of P'_n, P''_n, and M_n. This does, however, create a division by 0 if $P'_{n,x} - M_{n,x} = 0$, which is only the case for vertices that have identical x-coordinates as P_c, due to the curvature of the mirror. So if this happens, we assume the vertex is already optimally placed.

We found that the selection of vertex P_c is very important to the success of this approach. Our first criterion for a potential P_c is that its ray I_c should coincide with its reflection R_c to ensure that no deviation exists. This only requires some careful placement of V and all P_n, but also limits the ability to optimize the visibility of the reflective surface, as is done in [5]. However, with this single criterion, we have found perfect results for a 2D plane, as shown in Figure 6, but for 3D shapes this creates issues depending on which vertex is chosen. In Figure 7, all vertices in the indented area meet our criterion, but the normals are not completely restored. This is because all of these vertices are treated as equals, which leads to the reference vertex getting randomly chosen.



(a) 'Correct' model (b) Deformed model (c) Optimized model

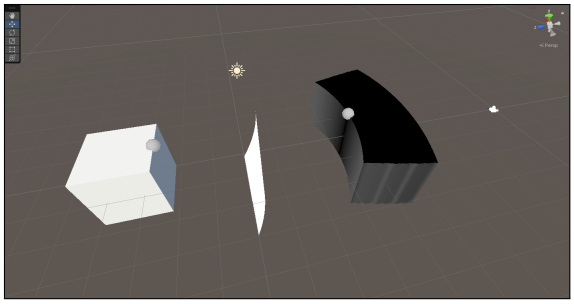
Figure 6: A flat plane distorted through a mirror has its normals restored with our 2D approach. Seen from mirror, deviation shaded.



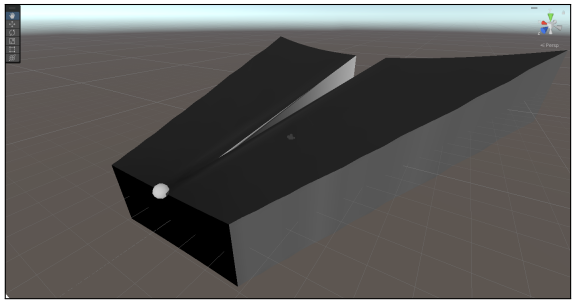
(a) 'Correct' (left) and deformed (right) model, with the mirror in between (b) Optimized model

Figure 7: A cube does not have its normal completely restored due to a sub-optimal choice of reference vertex (indicated by the gray dot on all models). Deformed and optimized model are deviation shaded.

We can solve the issue in Figure 7 by confining the P_c candidates to only the vertices that have normals mostly facing the mirror or the vertices closest to the mirror, but that creates other issues in our setup due to the divergence of the reflection rays, which is shown in Figure 8.



(a) 'Correct' (left) and deformed (right) model, with the mirror in between



(b) Optimized model

Figure 8: A cube with the reference vertex (indicated by the gray dot on all models) chosen from its front face diverges greatly in its back face. Deformed and optimized model are deviation shaded.

This divergence can be constrained by only optimizing the vertices visible in the mirror's reflection, but as we continued to experiment with this method by placing the camera in different positions, we realised it would not be a general solution. For example, the deformed object in Figure 7 completely blocks the view of the mirror from the viewpoint. This defeats the idea behind anamorphic art. Moving the camera to a different position, up along the y-axis for example, does make more of the mirror visible, and thereby the 'correct' version as seen in the mirror, but this also reveals any poorly optimized normals on the top of the model, as they would still not be properly placed.

2.2 Three-dimensional Approach

Our second method also uses a reference point as the base of its algorithm. After the initial deformation as described in [5], we choose one triangle face as the reference point, instead of a reference vertex like we did previously. This approach chooses the triangle whose normal faces the mirror the most as reference point. In our setup, this is detected by taking the triangle with the normal with the largest z-component.

Using the angle between the normal vector $\overline{N}_t$ of a triangle face connected to our reference face as well as the normal $\overline{N}_r$ of the reference face, we can create a plane that the new triangle face's vertices should lay in. Intersecting this plane with the reflection ray of the triangle's vertices would allow us to place the missing vertices.

However, at some point a triangle will have to placed when its vertices have already been positioned, like the sixth triangle in Figure 9. In the figure, all vertices of triangle 6 have already been placed, but the triangle is not yet defined at this point. However, it is impossible to position the plane of triangle 6 such that is rotated correctly relative to triangle 4 and triangle 5, precisely because all the vertices of triangle 6 have already been placed.

Another problem that became clear after implementing this approach is in the case of a triangle face that lies almost parallel to a reflection ray. This causes the intersection point of the triangle's plane and the reflection rays of the vertices of the triangle to be far removed from the center of the model.

Overall, this method performed worse than our 2D approach, as it is inherently unable

to reconstruct a mesh, such as the sphere shown in Figure 10.

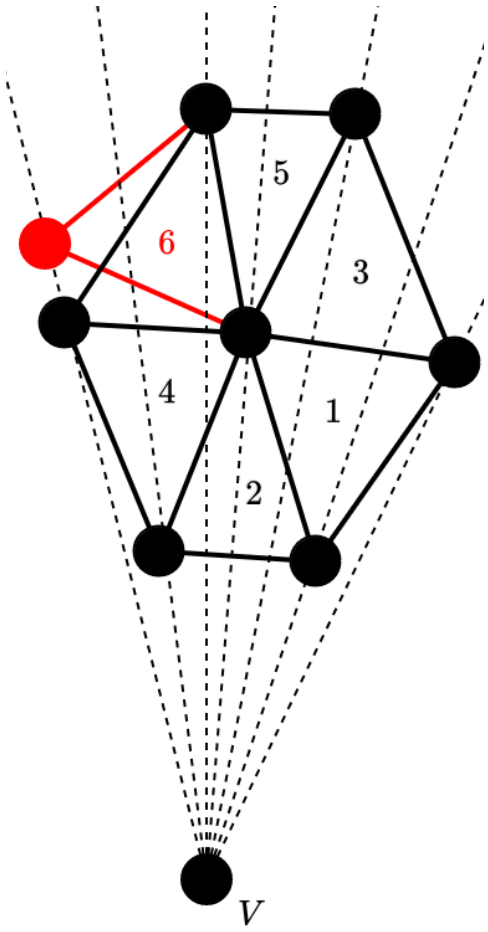
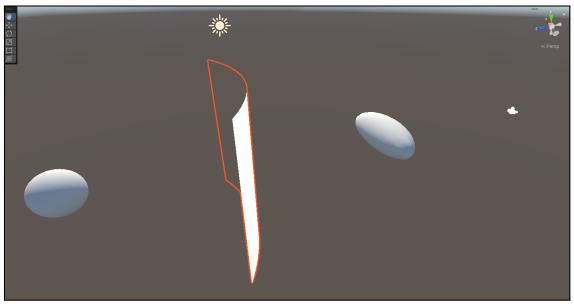
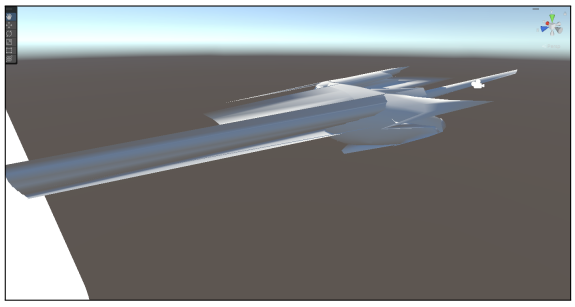


Figure 9: Triangle 1 as the reference face. The 3D approach then places triangles 2 through 5, in order, by using the dashed reflection lines to place each face's vertices. Instead of connecting the vertices of triangle 6, the algorithm places the sixth triangle relative to triangle 5, and the intersection of the missing vertex with its reflection ray is off from its correct point.



(a) 'Correct' (left) and deformed (right) model, with the mirror in between



(b) Optimized model

Figure 10: A sphere is poorly optimized with our 3D approach. Phong shaded.

2.3 Approximate Solution

With the restrictive nature of the first method and the errors inherent in the second, we moved on to our final approach. This method attempts to minimize the total angular deviation of the normals of the model by approximation.

For initialization, we again apply the anamorphic deformation method described in [5]. We then calculate the angular deviation of the normal of each vertex of the deformed object relative to the 'correct' model. These are summed to obtain the total angular deviation α_t of the model in its deformed state at $t = 0$. This is equivalent to $\alpha_t = \sum_{i=0}^{n} \arccos\left(\frac{\overline{N}_{d,i} \cdot \overline{N}_{c,i}}{|\overline{N}_{d,i}| \cdot |\overline{N}_{c,i}|}\right)$, where $\overline{N}_{d,i}$ is the deformed normal for vertex i, and $\overline{N}_{c,i}$ is its 'correct' normal.

Each iteration afterwards, we choose a random vertex and displace it a small random distance $d_d \in [-0.1, 0.1]$ along its reflection ray. We calculate the normals for this proposed change as well as the new total angular deviation α_t at time $t > 0$, and compare it to the previous α_{t-1}. If $\alpha_t \leq \alpha_{t-1}$, we accept the new position. In the other case, $\alpha_t > \alpha_{t-1}$, we usually reject the change. However, to prevent our algorithm from getting stuck in a local minimum, we use simulated annealing [7] to allow the algorithm a chance at accepting this 'bad' change. This acceptance chance is calculated as $\exp(\frac{\alpha_t - \alpha_{t-1}}{T})$, where the temperature T gradually decreases to 0.00, stopping at 0.01 to prevent division by 0 in the final iteration. If the acceptance probability is higher than a random number uniformly chosen from $[0, 1]$, the 'bad' change is accepted. As T decreases, this probability will gradually become smaller, which will generally ensure the algorithm ends in the global minimum.

As we show in Section 3, the random vertex and displacement selection forms bumps and spikes on the mesh. In order to reduce this, we also experiment with a smoothing pass after the optimization iterations are finished. We use a Laplacian smoothing function in this pass. This algorithm places the optimized vertex at the average location of its k neighbours in the mesh. However, this new position might not be on the reflection ray, which is the main constraint in the deformation algorithm and the cornerstone of the anamorphic reflection. To solve this, we project this smoothed point back onto its reflection ray by taking the closest point on the ray.

During our testing we found that completely using this smoothed location is not beneficial. We therefore reduce the influence of the smoothing algorithm by linearly interpolating between the smoothed location and the original optimized location. We perform this interpolation prior to the projection to the reflection ray.

As we use Unity, we had to make some concessions on execution time. Unity performs operations every frame. Instead of executing all iterations in one frame, we perform one iteration each frame. While this slows down the overall execution time, its main benefit is the prevention of frequent crashes. If we did not implement it this way, these crashes would happen once the Unity Editor UI would update, which happens on a mouse movement. Overall, this means that the time taken for all iterations is dependent on the rendering speed of each frame.

3 Results and Discussion

We test our approximate method with the same objects as we did the rejected methods: a plane, a cube, and a sphere. For each object, we show the 'correct' object, deformed object, and optimized object. We also plot the total angular deviation of the normals (TADoN) over time. Each optimization is done with 100,000 iterations, performed five times to obtain averages for each metric. The temperature of each optimization starts at 1.00, and is linearly decreased to 0.01 over the 100,000 iterations.

The optimizations were performed on a 2019 MacBook Pro running MacOS 14.3.1, with a 2.3 GHz 8-Core Intel Core i9 processor and 16 GB of 2400 MHz DDR4 RAM. Unity uses the machine's 4GB Radeon Pro 560X discrete graphics card. Apart from the total angular deviation, we also record how long each optimization took to perform on this machine in real time. It should be noted, however, that the real time taken is dependent on the rendering speed of each frame, as mentioned in Section 2.3. The results for each shape are shown in Table 1.

Object	**Vertices**	**Iterations**	**Time (ms)**	**Deformed TADoN**	**Optimized TADoN**
Plane	961	100,000	1,655,029	9666.56	6796.08
Plane	961	500,000	8,475,589	9666.56	5611.88
Cube	1734	100,000	1,475,097	11270.56	7894.66
Sphere	515	100,000	1,380,971	6416.68	3100.85

Table 1: Average performance metric of approximate approach.

In Figure 11 the algorithm is applied to a plane, for 100,000 iterations. As the figure shows, the total angular deviation was reduced, but it did not completely restore the relative normals and even introduced bumps. Overall, the TADoN for the plane was reduced by 30% with optimization, in 28 minutes. Interestingly, this optimization introduced vertical lines of almost optimal normals in the plane, despite the random selection of vertices to displace. The random selection and displacement also caused bumps to form on the surface of the plane. This is a behaviour that will also be seen in the other two shapes.

We suspected that the vertical lines could be reduced by more iterations, and to test this, we performed a single optimization of 500,000 iterations on the plane, taking almost 2.5 hours. However, as Figure 12 shows, not only were the vertical lines not reduced, the bumps got worse and became spikes. The five times longer optimization also did not reduce the TADoN by much more than with 100,000 iterations. It appears there is a balance to be struck between optimization iterations and the temperature decrease. Perhaps an initially faster temperature decrease could help prevent this spike formation and produce a lower TADoN.

Running the algorithm for 100,000 iterations on the cube (Figure 13) shows a decrease in TADoN of 30%. However, despite having such a decrease, none of the sides show significant improvement. The front's dark portion in the middle is slightly wider, but not as much as the plane. The sides and top are also disturbed, even though they were not visible in the mirror, as our setup (Figure 3) only looks at the front face of objects. This can likely be solved

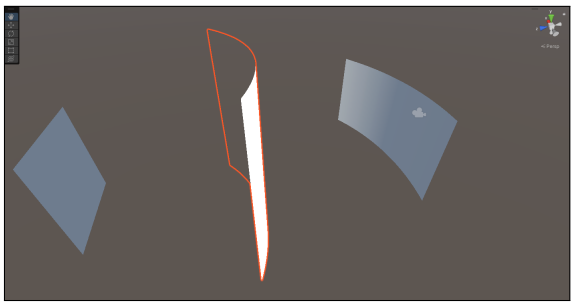
Object	Plane	Cube	Sphere
TADoN 10% **smoothing**	6828.68	8015.45	3854.27
TADoN 25% **smoothing**	7049.39	8482.08	4737.26
TADoN 50% **smoothing**	7552.61	9377.28	6443.63
TADoN 100% **smoothing**	8571.11	11365.48	7121.54

Table 2: Average performance metrics of approximate approach with smoothing of various strengths at 100,000 iterations.

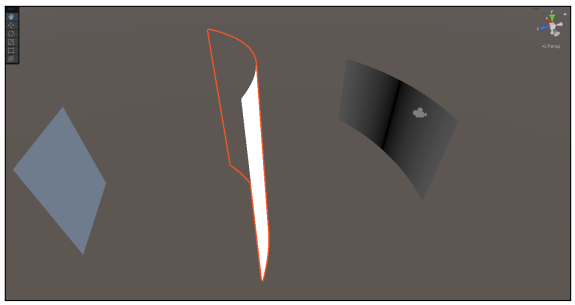
by only optimizing the vertices that are visible through the mirror, and leaving the other vertices as they are. The optimization was slightly shorter than the plane at 25 minutes, despite having almost double the vertices.

The last model we test the approximation method on is a sphere. As Figure 14 shows, the normals are much improved from the front, although there are some bumps. However, issues arise when looking from the side or top. It seems that large spikes have formed at those locations, similar to what happened with the larger amount of iterations on the plane. We believe this to be due to the same cause: a combination of random vertex selection, random changes along the reflection ray, and a high enough temperature have led bad changes often getting accepted for some vertices at the sides of the mesh. Overall, the sphere's TADoN was decreased by 52%, over 23 minutes.

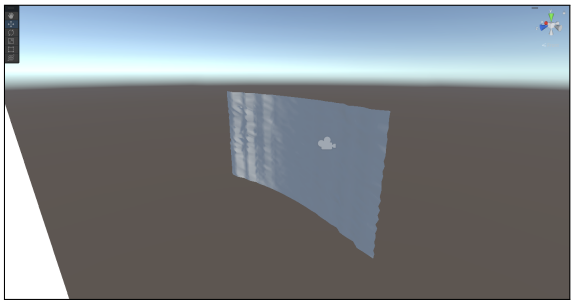
To remove the bumps and spikes in all of the optimized models we can smooth the optimized mesh, using the method described in Section 2. After doing so, however, it did not have much effect on any model. In all cases, smoothing the optimized model caused an increase in the TADoN (Table 2). We were unable to implement a way to see the deformed model through the mirror to view the full anamorphic effect, but Figure 15 shows the effect of smoothing on the side of the sphere that would be visible in the mirror. Lowering the smoothing strength does cause less of an increase, but any bumps also stay more pronounced, and some bumps around the edges were enlarged. The edges of the model also become worse when looking from the mirror. The divots seen on the top left and bottom right become worse with each smoothing increase, disrupting the illusion.



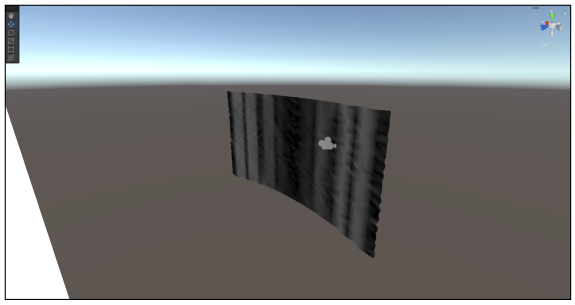
(a) 'Correct' (left) and deformed (right) model, with the mirror in between, Phong shaded



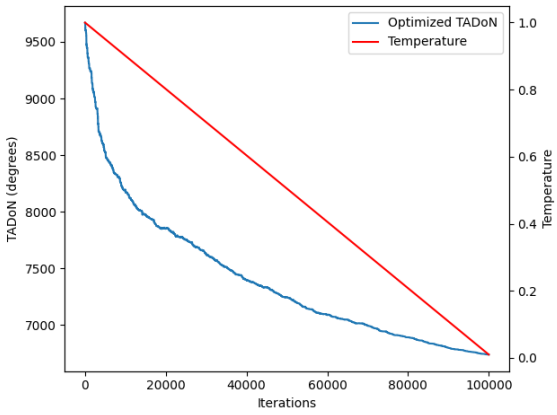
(b) 'Correct' (left) and deformed (right) model, with the mirror in between, deviation shaded



(c) Optimized model, Phong shaded

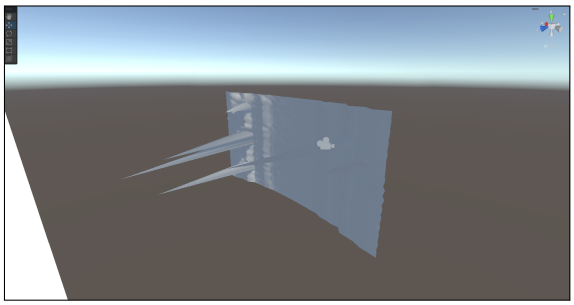


(d) Optimized model, deviation shaded

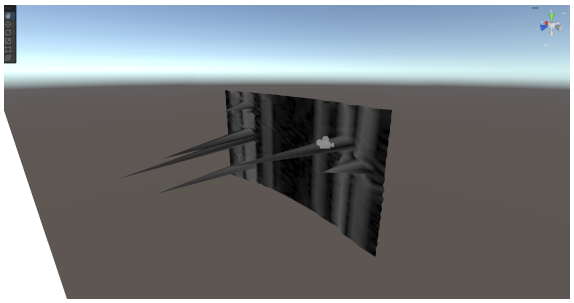


(e) Decrease in TADoN and temperature over time

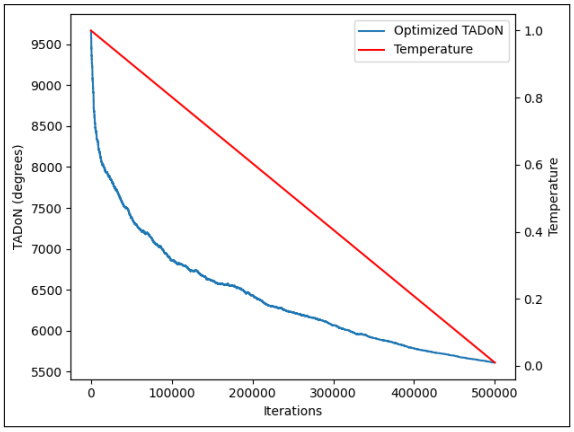
Figure 11: A plane with 961 vertices that is curved due to the reflection of the mirror. Iterative minimization of TADoN for 100,000 iterations did not fully restore normals.



(a) Optimized model, Phong shaded

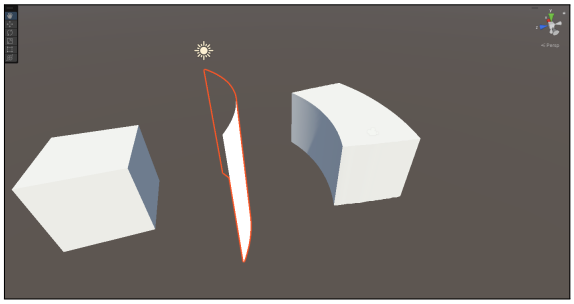


(b) Optimized model, deviation shaded

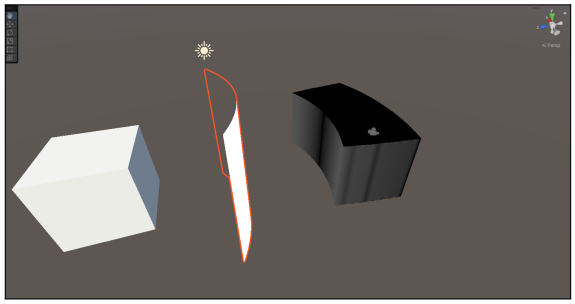


(c) Decrease in TADoN and temperature over time

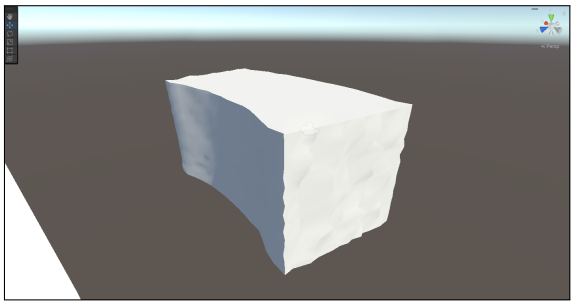
Figure 12: The plane from Figure 11, but optimized for 500,000 iterations.



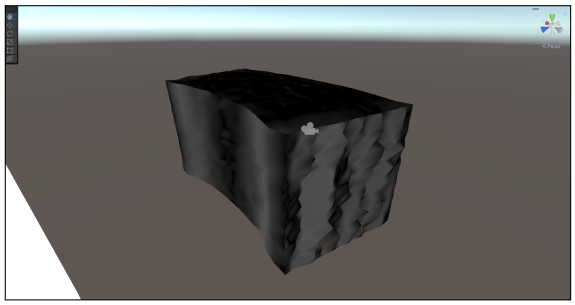
(a) 'Correct' (left) and deformed (right) model, with the mirror in between, Phong shaded



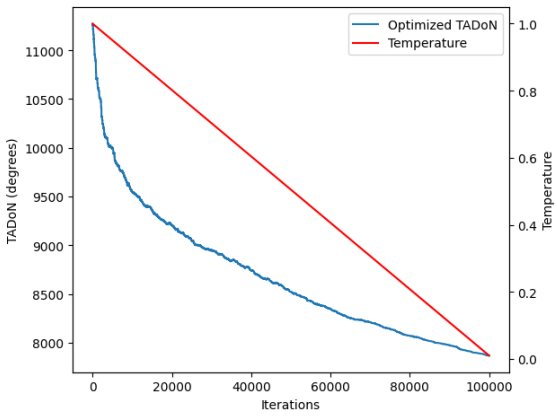
(b) 'Correct' (left) and deformed (right) model, with the mirror in between, deviation shaded



(c) Optimized model, Phong shaded

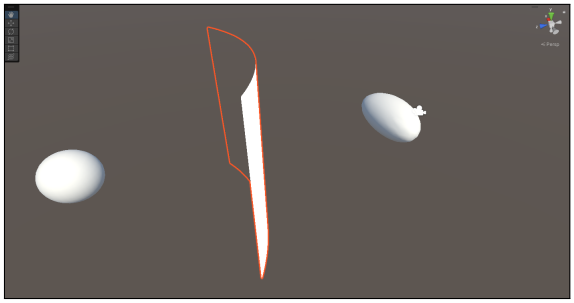


(d) Optimized model, deviation shaded

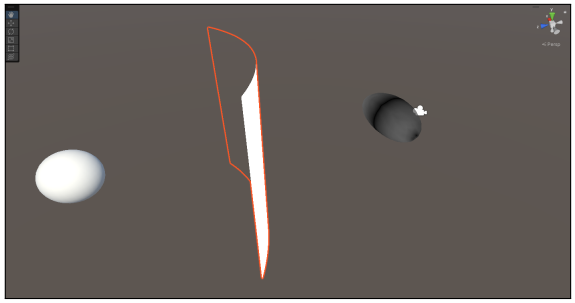


(e) Decrease in TADoN and temperature over time

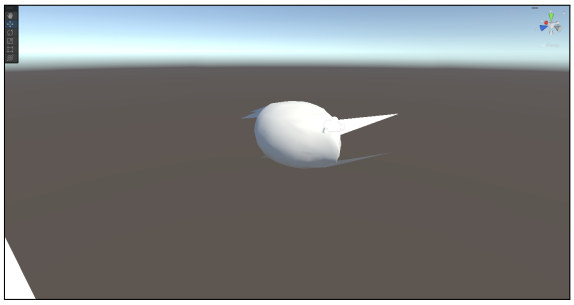
Figure 13: A cube with 1734 vertices that is curved due to the reflection of the mirror. Iterative minimization of TADoN for 100,000 iterations did not fully restore normals.



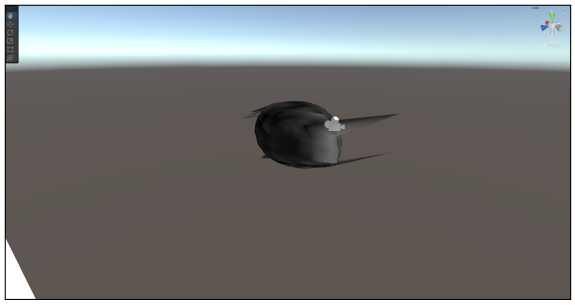
(a) 'Correct' (left) and deformed (right) model, with the mirror in between, Phong shaded



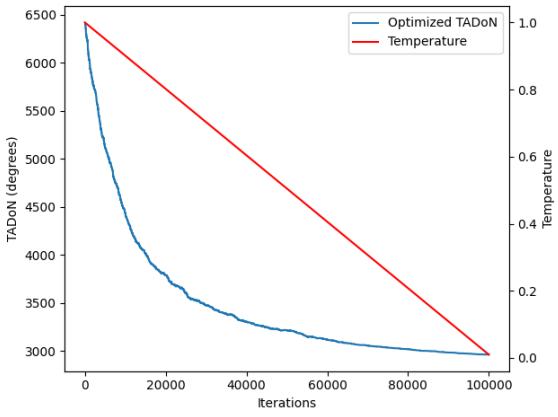
(b) 'Correct' (left) and deformed (right) model, with the mirror in between, deviation shaded



(c) Optimized model, Phong shaded

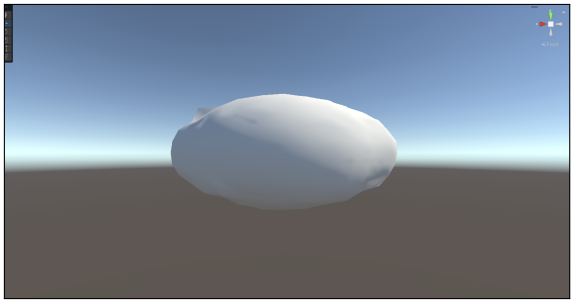


(d) Optimized model, deviation shaded

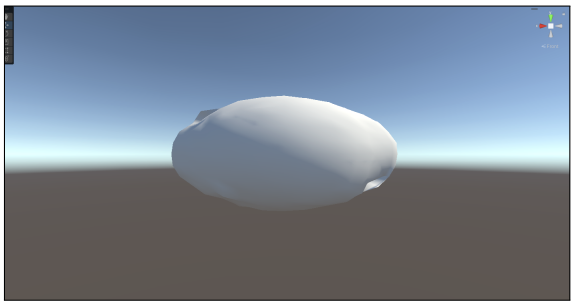


(e) Decrease in TADoN and temperature over time

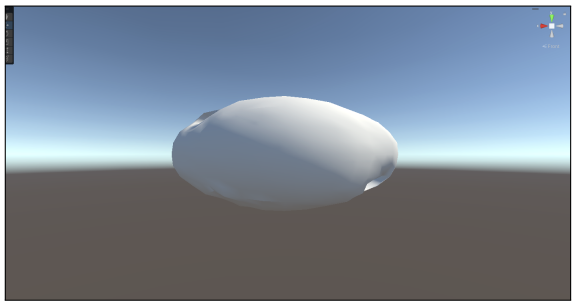
Figure 14: A sphere with 515 vertices that is curved due to the reflection of the mirror. Iterative minimization of TADoN for 100,000 iterations seed 0 did not fully restore normals.



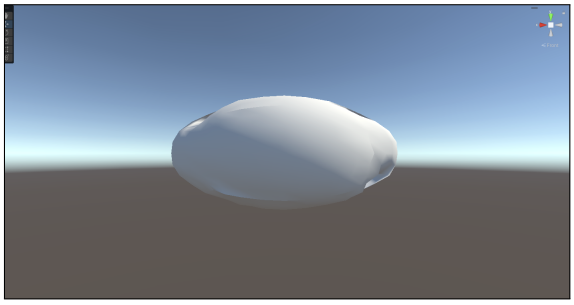
(a) No smoothing



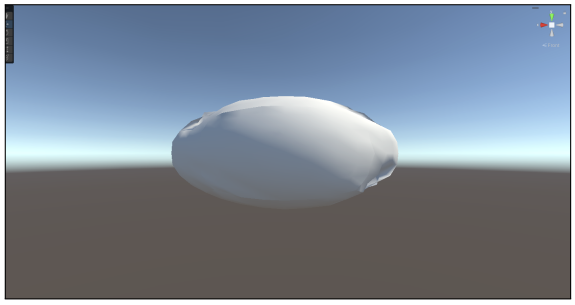
(b) 10% Smoothing



(c) 25% Smoothing



(d) 50% Smoothing



(e) 100% Smoothing

Figure 15: Laplacian smoothing applied to the optimized sphere of Figure 14. Only showing the side visible from the mirror.

4 Conclusion

We have experimented with various methods to restore the relative normals of an object that has been deformed through a mirror. Our methods include an exact two-dimensional approach, an exact three-dimensional approach, and an approximation using simulated annealing. We tested these methods in a scene with a single convexly curved mirror and a light placed at infinity. Only the diffuse lighting component is considered.

The exact two-dimensional approach (Section 2.1) performs perfectly on a flat plane, but quickly breaks down on objects with more depth due to issues with the reference vertex selection. This selection could be made more strict, but there are other issues with this method. For example, the method only optimizes the side of the deformed object facing the mirror. This approach works if this face is the only visible face, but requires the deformed object to be placed between the viewpoint and the mirror, blocking the mirror and invalidating the artwork. Moving the viewpoint so that the mirror is not blocked by the deformed object reveals other sides of the object. These sides are not optimized by our two-dimensional approach.

Our three-dimensional approach (Section 2.2) uses a reference triangle face, instead of a reference vertex. Placing neighbouring triangle faces by intersecting their planes with the reflection rays of the vertices making up the neighbouring triangle, should have created a deformed model with perfect relative normals. However, due to an eventual lack of freedoms to place the vertices of a neighbouring triangle face, this method is wholly unable to create a coherent model.

Our final approach (Section 2.3) approximates the optimal deformed mesh by minimizing the total angular deviation of normals. That is, the sum of the angle between the 'correct' normal and deformed normal for each vertex of the object's mesh. We applied this technique to a flat plane, a cube, and a sphere for 100,000 iterations. We found that our method is able to significantly reduce the total angular deviation. We saw reductions of 30% $\sim$ 50%, depending on the model. However, the algorithm creates a bumpy surface and takes significant amounts of time. For a very simple mesh of 515 vertices, the algorithm took 23 minutes. The objects with slightly more vertices took up to five minutes longer, but due to the plane (961 vertices) taking longer than the cube (1734 vertices), despite having fewer vertices, we cannot derive a relationship between vertex count and time taken for optimization of a certain amount of iterations. Objects that represent more complex shapes are made up of thousands of vertices, which will increase the time even further.

Performing the approximation for more iterations did not significantly reduce the TADoN approximation metric. Quintupling the number of iterations only led to an additional decrease of 10%. Due to the slower decrease in temperature, the acceptance probability remained high for a longer time. This led to more 'bad' vertex positions getting accepted, leading to spike formation. Additionally, these extra iterations changed the time taken from 28 minutes to 2 hours and 21 minutes.

In a final attempt to reduce the bump and spike formation, we applied a Laplacian smoothing pass at various strengths. Although smoothing the mesh after approximation led to an increase in TADoN, it did lessen bumps and spikes, as expected. However, at 100% smoothing strength, the cube and sphere saw a smoothed TADoN that was higher than in its deformed stage, which is not desirable.

5 Future Work

Our research has shown the possibility of restoring the normals of an object that has been deformed by a reflective surface, as is the case in anamorphic sculptures, but the results of such optimizations leave many flaws that are yet to be resolved.

Our approximation approach creates bumps due to the random selection of vertices and offset values. We extended our method with a smoothing pass, but saw little positive results from it. One avenue of investigation could be alternating the iterative approximation with smoothing passes at various strength. The approximation phase would reduce the TADoN, after which a smoothing pass removes any bumps that have formed. Alternating this process for multiple iterations could lead to improved results.

Another method to try to limit spike formation could be to constrain the vertices to remain within a certain distance of its neighbours, based on the original mesh's proportions. This could stop the formation of the large spikes we saw in the approximation of 500,000 iterations.

Alternatively, experimenting with non-linear annealing speeds could also reduce spike formation due to an earlier lower temperature and consequently lower acceptance probability.

Additionally, we were only able to test our algorithms on relatively small meshes due to time and technical constraints. The iterative approximation takes significant amounts of time, and our machine was already heavily taxed running the small meshes. In the future, we would like to determine various relations between the metrics of our optimization by applying our technique to a larger number of more complex models and running it for longer. Relations such as the number of iterations and decrease in TADoN, or the number of iterations, vertices, and duration of the algorithm could be used to determine if there is a limit to what our algorithm can reasonably optimize.

The implementation of our algorithm can also be streamlined, which could greatly reduce the time taken on each iteration as well as the effectiveness of the algorithm as a whole. One such optimization would be to optimize only those vertices that are visible in the reflective or refractive surface. In the case of the cube we tested with and the scene we tested it in (Figure 3), this would reduce the amount of vertices that are considered for optimization by $\sim$83%, as five of the six sides are not visible from the mirror.

Lastly, we have two ideas for alternative methods to solve the issue that we were unable to experiment with due to time constraints. The first is that, instead of modifying the mesh, perhaps modifying the lighting itself could prove useful. We currently have one directional light placed at infinity, but maybe playing with other types of lighting only for the deformed model could create different results.

The second alternate method is to colour the deviating parts of the deformed model. Parts that are too light could be painted in a darker colour, and areas that are too dark could be painted in a lighter colour. This does require that the base object is not an extremely light or dark colour, in which case it will be impossible to make areas lighter or darker by colouring the object itself.

Acknowledgements

I would like to thank Jiří Kosinka and Steffen D. Frey for their support and guidance during this research internship. It was an interesting project and a great experience.

References

[1] M. Navoni and F. Buzzi. *Leonardo Da Vinci and the Secrets of the Codex Atlanticus*. White Star Publishers, 2019. ISBN 9788854413528. URL `https://books.google.com.au/books?id=EeNzuwEACAAJ`.

[2] Agostino De Rosa and Alessio Bortot. *Anamorphosis: Between Perspective and Catoptrics*, pages 243–289. Springer International Publishing, Cham, 2021. ISBN 978-3-319-57072-3. doi:10.1007/978-3-319-57072-3_38.

[3] Francesco de Comite. A General Procedure for the Construction of Mirror Anamorphoses. In *Bridges 2010: Mathematics, Music, Art, Architecture, Culture*, pages 231–238, Pecs, Hungary, July 2010. URL `https://hal.science/hal-00861388`.

[4] Francesco de Comite. A New Kind of Three-Dimensional Anamorphosis. In *Bridges 2011: Mathematics, Music, Art, Architecture, Culture*, pages 33–38, Coimbra, Portugal, 2011. URL `https://hal.science/hal-00861392`.

[5] Louis Pratt, Andrew Johnston, and Nico Pietroni. Bending the light: Next generation anamorphic sculptures. *Computers & Graphics*, 114:210–218, 2023. ISSN 0097-8493. doi:https://doi.org/10.1016/j.cag.2023.05.023.

[6] Matt Pharr, Wenzel Jakob, and Greg Humphreys. *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann Publishers Inc., 2016. ISBN 9780128006450.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. doi:10.1126/science.220.4598.671.