Estimating image distortions for mirror anamorphoses using sampled point displacements

Author: Gees Brouwer
Supervisors: Baran Usta, Prof. Dr. Elmar Eisemann

June 27, 2021

Abstract

The principle of a *mirror anamorphosis* relies on distortion caused by a reflective object of a particular shape and the perspective of a viewer looking into the reflective object to look at an image on the surface. The distortion of the reflected image is intended to form a recognizable image, while the image on the surface looks completely different.
There are many ways a computer could aid the creation of this form of art by determining how the image is being distorted, but in most cases a regular artist would not know specific geometric details of the scene. In these cases a solution is for the creator to provide any image on the surface and an image of its distortion caused by the scene. The creator can match points between both images to guide the computer into calculating the (estimated) distortion.

Keywords: Mirror anamorphosis, distortion correction, interpolation

1 Introduction

Artists' experimentation with perspective techniques and distortion lead to interesting types of art throughout history [BAL77]. The principle of *anamorphosis* in art relies on distortion caused by the way an artwork is presented to the viewer. For example, many types of anamorphic street art utilise *perspective distortion* in order to make a distorted two-dimensional drawing appear like a deep abyss, from the right point of view. A mirror can be used for taking this principle a step further, where a distorted image needs to be reflected in a reflective object of a particular shape before being viewed; this is called *mirror(catoptric) anamorphosis*. While this principle offers a wide variety of distortions depending on the shape of the mirror, it can get extremely hard for artists to resolve the correct distortions to enable the illusion. This can be determined by a trial-and-error process or by mathematical calculations. Luckily computer science has a lot of potential to be a ’helping hand’ for artists in this form of art.

Computers can automate parts of creating mirror anamorphoses by computing the distortions, even for more complex reflective shapes.

The aim of this paper is to describe an approach to enable artists to directly design their artwork like how it will appear to a viewer, while a computer transforms the artwork to compensate for the distortion. Solutions like *ray tracing* [GLAS89] exist for calculating how to distort an image to simulate its reflection in a scene, but for this approach we assume the artist being unable or unwilling to capture the 3D data of their mirror anamorphic scene.

Instead, 2D data in the form of images will be used; the first one being *any* image, the second one being the way the first image will be perceived by a viewer of the mirror anamorphosis, as if the first image would be the artwork.

The approach will involve a user interface for the interactive creation of a mapping between the images. With minimum user assistance in mind, the user will select a few matching points. By knowing how these points are displaced, the displacement for every pixel of any image will be estimated by the computer; this will result in a full mapping that is able to simulate the distortion of an artwork for a mirror anamorphic scene, as well as an inverse mapping to be used for resolving the distortions for an artwork as accurately as possible.

Goal: *Calculating a mapping with minimum user assistance for resolving the distortion for an artwork used in a mirror anamorphosis.*

In theory, the approach could require the user to match every pixel of the two images to obtain a full mapping. Assuming no errors by the user are made, such a mapping would simulate the distortion perfectly at pixel-level and would be able to produce results of the same quality to *ray-tracing* [GLAS89]. Realistically, this would be too time-consuming. A compromise must be found between user effort and the potential of computer science, hence the formulation 'minimum user assistance'. This paper will therefore consider and compare *estimation methods* for estimating an accurate full mapping while using a *minimal mapping* created by the user as input. User interface design for iteratively creating such a minimal mapping, by using cycles of requesting input and review of the output, for stepwise improvement of the mapping will also be considered.

2 Related work

2.1 Texture mapping

The approach that will be described in this paper is similar to *texture mapping,* the method of mapping pixels from a flat texture image to a 3D model. Such a texture can be considered 'an artwork' that will be distorted due to the perspective and transformations in 3D space. The result of texture mapping and rendering a 3D model will be a planar projection of the 3D models' vertices onto *screen space,* with the texture being fitted in between. This can all be compared to the way a viewer perceives a mirror anamorphic artwork; a planar projection of an artwork that is distorted due to the perspective and transformations in 3D space caused by the shape of the mirror. Thus, it is no coincidence that this approach also tackles the problem of fitting an image through vertices with a corresponding texture coordinate. In *State of the art in Computer graphics: Visualization and Modeling,* Heckbert, P. S. [HEC91] describes existing methods for texture mapping; how *affine* texture mapping uses *linear interpolation* in screen space but introduces a 'rubber sheet effect' due to perspective-incorrectness; all other techniques perform calculation on the 3D data of the vertices for (higher) perspective-correctness.

This is where our approach and *texture mapping* part their ways. The lack of 3D data to consider depth will eliminate all perspective correct existing methods for texture mapping and calls for research into alternative algorithms for fitting the pixels of a texture in between vertices.

2.2 Multivariate interpolation

The approach that will be described in this paper will involve estimating the values of points in 2D space by using surrounding points with values that are given; the values in this case being *displacement vectors* of points. The established values will be irregularly placed, in the field of estimation methods known as '*scattered data'.* Estimation methods for values on a regular grid are significantly easier to perform, since many can be applied for each dimension separately and the 'closeness' to neighbouring values is easier to determine.

Cuomo, S., Galletti, A., Giunta, G., & Marcellino, L. [CGGM14] have described a method for 'extending' estimation methods for values on a regular grid to deal with scattered data by using triangulation and barycentric coordinates.

Amidror, I. [AMI02] has compared several *interpolation* methods for estimating data in two-dimensional and three-dimensional spaces in his work *Scattered data interpolation methods for electronic imaging systems: a survey.* He also compares the basic differences between *data interpolation* and *data fitting.*

An important difference between the two methods is the fact that interpolation will consider the established values as correct and distinct while fitting methods can deviate from one or more established values for the sake of finding a better fitting.

The approach will use the knowledge shared in these works to find a suitable estimation method.

2.3 User interface for sampling

The work [DUKR18] discusses user interface design for interactive machine learning by presenting intermediate (training) results and guiding the user into giving input that will be likely to improve the results. These iterative cycles of input and review inspire the way the interface for this approach will be built. Since this approach uses *estimation,* the correctness of the estimation will depend on the (amount of) user input. By presenting intermediate results to a user there can be decided if more sample points will be needed.

Subdivision surfaces as described in the work [SHAR00] inspire the way iterative cycles are carried out; by presenting the user's input as a polygonal model, the accuracy of the estimations within each polygon can be evaluated. Polygons without sufficient accuracy will request a sample point to subdivide the polygon. This process will repeat recursively until each polygon has sufficient accuracy.

3 Method

3.1 Purpose

The aim of this paper is to describe an approach to aid the creation of mirror anamorphoses, by calculating mappings. To understand the role and importance of such mappings, we first illustrate the general principle of mirror anamorphoses. Mirror anamorphoses generally consist of two objects:

- The artwork of the artist, e.g. a painting/drawing/picture, usually planar and horizontally placed on the surface
- A reflective object, strategically placed so that a viewer can see the entirety of the artwork being reflected

Mirror anamorphoses will show a viewer a recognizable image as a reflection using the reflective object; that means that the artwork itself will be distorted and the reflective object 'undistorts' the artwork, or in other words, the artwork must be transformed in such a way it will *compensate* for the distortion the reflective object will cause. In some cases, the distorted artwork can even be another recognizable image by itself [BAL77].
The goal of this approach is to enable artists to directly design their artwork like how it will appear to a viewer, while a computer transforms the artwork to compensate for the distortion. While other possibilities arise, e.g. 'previewing' the reflection of a manually distorted artwork, this will be the only goal that will be considered.
For a computer to transform any artwork in such a way it will *compensate* for the distortion the reflective object will cause, it will need to determine how to map any image to match its reflection perceived by a viewer.

*Suppose there is a mapping X and an image A such that X * A matches image B, an image of the reflection of A. (((()))) Then,* $X^{-1} * B$ *will match image A. Suppose we create a new image C defined by* $X^{-1} * A$*. Determining the reflection of C will give* $X * A = X (X^{-1} * A) = (X * X^{-1}) * A = A$.

We see that by calculating a mapping to simulate the distortion caused by the reflective object**,** we can apply the inverse of the mapping to compensate any image for the distortion the reflective object will cause.

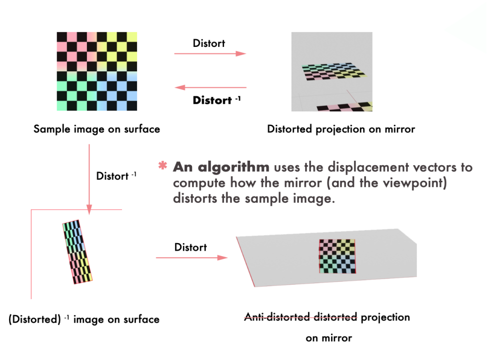

Figure 1: By 'anti-distorting' an image, indicated by 'distort^{-1}**'**, the distortion caused by the mirror will be compensated.

3.2 Input

For this approach we consider the artist being unable or unwilling to capture the 3D data of the scene. Therefore, calculating mappings based on the shape of the reflective object, e.g. by using *ray tracing* [GLAS89] is not possible. Instead we focus on a more straightforward approach for capturing data; this approach requires the artist to capture two sets of 2D data; a top view of the artwork and a planar projection of the reflection from the desired view point. An example could be two pictures taken with a smartphone camera.
An interface will be built for the artist to assist the creation of the mapping using the two images of the artwork and its reflection.
The interface will display the two images and will require its user to build a *partial* mapping by selecting a few matching points.

3.3 Output and interpretation of results

The approach aims to calculate the mappings as accurately as possible with minimum user assistance; the quality of the mappings will be taken into consideration instead of the time it takes to be calculated.

The approach produces an output by taking the inverse of the calculated mapping and transforming any image given as an input. As discussed in *3.1 Purpose,* using said output in the mirror anamorphic scene will result in the reflection resembling the image given as an input, depending on the quality of the mapping.

Said resemblance will be a direct result of the accuracy of the mapping and can therefore be used as a measure for obtaining results. The measure can be obtained by using methods for determining *image similarity.*

Said measuring requires (1) a user to capture the 2D data of the reflection for each output and (2) requires finding a region of interest for the captured 2D data and scaling for proper comparison with the original input image.

To avoid these problems we can produce a different output by (1) 'recycling' the two images used to create the mapping, (2) using the calculated non-inverted mapping to transform the first image in order to 'simulate' its reflection. This output can then be compared to the second image, being the actual reflection of the first image. Since both images already exist there is no need for the user to capture any data. Also, since the first image will be mapped to the same scale and position as the second image there is no need for finding a region of interest and considering scaling; the alternative output can be directly compared to the second image.

*Suppose the approach calculates a mapping X for a mirror anamorphic scene and suppose there is an 'actual' mapping Y for the scene, unknown to both the user and the computer. For the calculation of the mapping, image A and the image of its reflection B, defined by Y * A are used.*

*The alternative output for the approach is defined by X * A. Representing the image similarly method as 'subtraction', comparing the alternative output to image B will give: X * A - B = X * A - Y * A = X - Y. This shows that these methods will measure the direct result of the accuracy of the mapping.*

4 Results

4.1 Using linear interpolation

The first results will be obtained by using linear interpolation; [CGGM14] described a method for linearly interpolating on multidimensional scattered data; first a triangulation method will be performed on the data points to subdivide the domain into connected triangles. Then, the value for any point in the domain can be estimated by using *barycentric coordinates*. Note that this approach is the same as the method of *affine texture mapping.* While this type of texture mapping has known drawbacks related to perspective-correctness [HEC91], it's interesting to review how these drawbacks will affect the results.

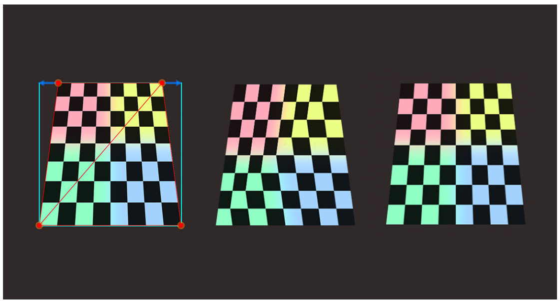

Figure 2: Left: The actual reflection, Middle: Result of using linear interpolation with triangulation, Right: Result of using linear interpolation with quadrilateralization

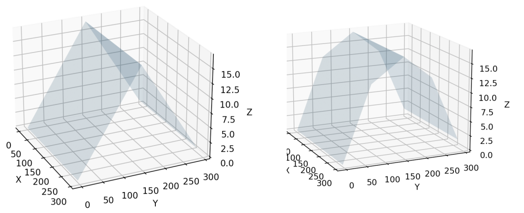

Figure 3: 3D plots of the estimation of the *vertical* displacements of each point in the domain using linear interpolation, the Z-axis represents the vertical displacement. Left: The 3D plot for 6 data points, Right: The 3D plot for 10 data points

4.2 Using cubic interpolation

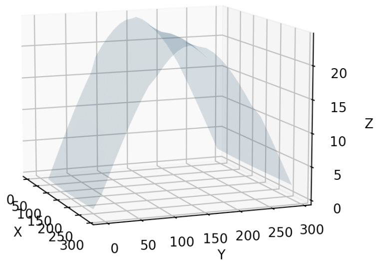

Figure 4: 3D plot of the estimation of the *vertical* displacements of each point in the domain using cubic interpolation, the Z-axis represents the vertical displacement.

4.3 Using RBF interpolation

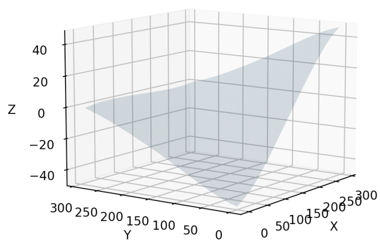

Figure 5: 3D plot of the estimation of the *horizontal* displacements of each point in the domain using RBF interpolation, the Z-axis represents the vertical displacement.

5 Discussion

As discussed in the section "Related works" the approach as described in this paper combines existing methods.

6 Conclusions and future work

In this paper we described an approach for calculating a mapping with minimum user assistance for resolving the distortion for an artwork used in a mirror anamorphosis. For achieving the most accurate mappings while keeping the user assistance minimal we compared several estimation methods. We compared the differences between *data fitting* and *data interpolation* and there were no results needed to eliminate any data fitting method; the only benefit of *data fitting* over *data interpolation,* being the possibility of correcting user errors, could be resolved for any data interpolation method by user interface design to keep user errors minimal.
The first results by using linear interpolation showed why the interpolation method should use more than just the surrounding data points, otherwise the transition between polygons will be visible for curves and perspective.

References

[BAL77] Baltrusaitis, J. (1977). Anamorphic art. Translated by WJ Strachan. Cambridge, England: Chadwyck-Healey.

[HEC91] Heckbert, P. S., & Moreton, H. P. (1991). Interpolation for polygon texture mapping and shading. *State of the art in Computer graphics: Visualization and Modeling*, 101-111.

[HEC83] Heckbert, P. S. (1983). *Texture mapping polygons in perspective* (Vol. 13). Technical Memo.

[SMI80] Smith, A. R. (1980, July). ‘Incremental Rendering of Textures in Perspective. In *SIGGRAPH’80 Animation Graphics seminar notes*.

[AMI02] Amidror, I. (2002). Scattered data interpolation methods for electronic imaging systems: a survey. *Journal of electronic imaging*, *11*(2), 157-176.

[CGGM14] Cuomo, S., Galletti, A., Giunta, G., & Marcellino, L. (2014). A novel triangle-based method for scattered data interpolation. *Applied Mathematical Sciences*, *8*(134), 6717-6724.

[DUKR18] Dudley, J. J., & Kristensson, P. O. (2018). A review of user interface design for interactive machine learning. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *8*(2), 1-37.

[SHAR00] Sharp, B. (2000). Subdivision surface theory. *Game Developer*, *7*(1), 34-42.

[GLAS89] Glassner, A. S. (Ed.). (1989). *An introduction to ray tracing*. Morgan Kaufmann.