Boundary Focused Graphics

Literature Review #1 - COMP.5460

Adam Gaudreau

28 February 2018

BOUNDARIES AND SHAPE INTERPOLATION

Shape interpolation is one of the fundamental subjects regarding computer graphics. There are many interpolation methods and techniques in practice, but they all aim to solve the same problem. To understand this problem, it helps to know what interpolation is itself.

Interpolation in computer graphics is the ”filling-in” of the matter between two frames. For example, imagine an image/frame of a plain stick figure with its hands by its side, and let’s call it the *source*. To its right is another image/frame of the same stick figure, the only difference being that its right hand is up by its head as if it’s waving to you. This is the *target*. If we want to make a 60fps animation of the *source* to the *target* with a duration of one second, we clearly need 60 frames. Given we have the first and the last frame, how do we know what the middle 58 frames need to be to have the animation appear smooth? Shape interpolation techniques all uniquely solve this problem, each having their own quirks.

Knowing what shape interpolation is, it’s easy to see how it’s important to many aspects of the modern computing experience. However, a problem arises in nearly every one of these shape interpolation techniques. Until now, planar shape interpolation (that is, shape interpolation in terms of a mathematical plane) has produced an output with no boundaries with respect to its input.[1] In simpler terms, the interpolation method will transform an object without boundaries, meaning that it is possible to have the image fold over on itself or have it produce undesired results. This makes the interpolation of the image seem qualitatively wrong to the viewer. The new method as described in ”Planar shape interpolation with bounded distortion” adds boundaries to an image’s interpolation so its distortion is controlled, unlike all previous methods.

To simplify the paper’s contributor’s research, these bounded distortions are achievable by comparing the input maps between the shapes rather than the shapes themselves. More specifically, they try to blend the input maps so they can ”agree upon” a certain boundary, if you will. This will help prevent large, unexpected amounts of distortion so that we can better qualitatively analyze the interpolations.

Again, I’ll describe another scenario to help visualize the problem at hand. Imagine an image of a cat curled up taking a nap. If we want the cat to wake up and uncurl itself to a standing position, we as humans can understand how that image needs to transform for it to do so. A computer obviously doesn't know this, so all it knows is that it needs to get from this frame to the next. Without boundaries on distortion, the computer may calculate the shortest way is to have the cat continue to curl up, folding over itself, to stand up. Visually to a human this would look unsettling since it's an unnatural (and impossible) movement in the real world. Chen et al. found a way to bound these distortions so the cat can move in an expected manner.

BOUNDARIES AND IMAGE FLATTENING

Boundaries aren't important for only the 2d plane; it also extends into the three dimensional plane. The 3d plane, however, has more characteristics that need attention. For instance, in the 2d plane, we can interpolate an image by rotating, translating, scaling reflecting, etc. To interpolate a 3d image though, we now have to consider and preserve the angles of that image proportionally. If we wanted to double the size of a 3d image of a human face, for example, we would have to make sure all the angles and curvatures of that face scale up properly. This is where *conformal flattening* comes in handy. Conformal flattening will represent a 3d image on the plane by "flattening" it, and it does so in a way that preserves the angles with no distortion. These flattened interpretations are called maps, and they are fundamental to many aspects of computer graphics.

Up until recently, there was a restricting design flaw in the algorithms for conformal flattening: there was no explicit control of the resulting shape. In other words, when the algorithm produced a result, it stayed as it is (take it or leave it). There exist methods that can provide some control over the resulting shape, but they run in non-linear time and are very resource consuming. Rohan Sawhney and Keenan Crane of Carnegie Mellon University came up with a method called *Boundary First Flattening (BFF)* that grants full control of the map in linear time.[2] Some of these controls include preservation of sharp corners, direct manipulation of boundary or angle, and mapping to a given target shape. One of the most impressive results of BFF is the ability to edit in real time with fast optimization of high-resolution maps.

To summarize the paper's in-depth algorithm design, BFF is achievable by finding the boundary data for the conformal mapping and applying it to the rest of the domain. In the best case scenario, this method is about 50 times faster than other methods.[2]

WHY BOUNDARIES ARE IMPORTANT

From the two aforementioned articles, it's easy to see how the consideration of boundaries in image processing and other graphical methods are beneficial. Not only can they provide more powerful and efficient methods, but they can also make image interpolation more natural to the human eye.

From the first article about shape interpolation, we saw how adding boundaries on distortion on the input will alter the output more naturally without undesired effects like fold-overs. This type of approach was taken in the creation of BFF. By focusing on the constraints around boundaries, the maps will be created while maintaining sharp angles, corners, and the correct amount of distortion.

It's important to consider the most preliminary concepts in graphics when trying to enhance today's methods and algorithms, such as boundaries. When we begin learning computer graphics, we focus on primitive shapes like lines, circles, and individual pixels. When we construct primitives with other primitives, the most important aspect about them is the boundary. For instance, when we are asked to render a rectangle on the screen, we need to establish the boundaries first. The incredibly more advanced methods used in today's graphics still keep these simple rules in mind to optimize existing solutions. These two articles, for instance, benefited from doing just that.

To conclude, boundaries are an essential aspect to any graphical method and/or algorithm and should be taken into consideration when trying to optimize and enhance existing and future solutions. In doing so, it may yield in better, more consistent results (if not, a way to distinguish whether a result is good or bad), and improve run times 50 fold.

Bibliography

[1] R. Chen, O. Weber, D. Keren, and M. Ben-Chen, "Planar shape interpolation with bounded distortion," *ACM Trans. Graph.*, vol. 32, pp. 108:1–108:12, July 2013.

[2] R. Sawhney and K. Crane, "Boundary first flattening," *ACM Trans. Graph.*, vol. 37, pp. 5:1–5:14, Dec. 2017.