Mimicking the Infinite Zoom Movie-making Effect

Luke Darcy
Carnegie Mellon Univeristy
1320 Wightman Street, Pittsburgh PA 15217
ldarcy@andrew.cmu.edu

Abstract

The infinite zoom effect is a niche yet interesting movie-making tool that is occasionally used for transitions or opening sequences. Creating this sequence involves the use of multiple different lenses, and a convoluted editing process to create a compelling result. I propose some new approaches and algorithms to simplify this process, and create this effect without extensive editing. I test the use of a wide angle lens and a single lenses and observe how this impacts the effect.

1. Introduction

There are many movie-making effects commonly used in the industry, such as the arc shot, the bird's eye shot, and the Hitchcock shot. This final shot involves moving towards an object while zooming in the opposite direction, and partially motivated me into mimcing movie-making effects.

Current methods for the infinite zoom effect call for professional cameras and complex editing sequences tailored for that specific scene. This has the potential to be both a resource and time drain for film-makers and movie studios.

My approach is designed to create the infinite zoom effect using an algorithm and a stack of photos that steadily "zoom in" on a point in the distance. This will minimize the time film-makers spend on re-creating this effect for each scene that calls for it.

The idea behind our algorithm is that a user would take a series of images such that each image is further into a scene, and the center of these images would all need to the same point, ideally far off in the distance. This algorithm is designed for long, straight scenes like a road, as the infinite zoom effect would require a deep scene to "zoom" into.

Throughout this paper I describe the various approaches I take to recreating this problem and finding an algorithm that best creates this effect. I explore the use of a wide angle and medium angle lens against just a medium angle lens, and I try a variety of small techniques that movie-makers use in conjunction with the infinite zoom effect to make their scenes more compelling.

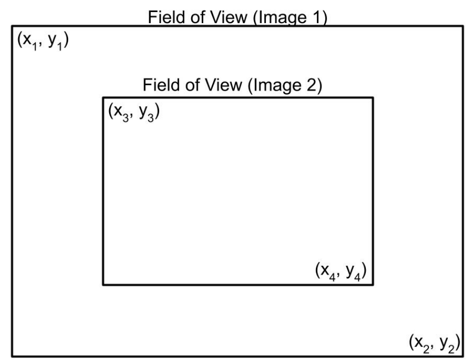

Figure 1. Demonstration of field of view on an image (image 1), and another image taken some depth into a scene (image 2)

2. Methods

In order to seamlessly move between two image sequences, I need to define a function that allows me to mimic zooming on any portion of an image. Our image stack will contain a sequence of images such that each image moves further into the scene. Figure 1 gives us an idea of the field of view of image 2 (which is deeper into the scene), and in order to move seamlessly in between these two images, I need to crop the first image such that these cropped images transition between the two fields of view. This is done by first defining a constant t where a value of 0 represents the field of view of the first image, value of 1 represents the field of view of the second image, and any value in between 0 and 1 represents represents a field of view rectangle that is in between the other two field of view rectangles. I also define the upper left corner of the transition rectangle with this formula:

$$(x_1 + t * (x_3 - x_1), y_1 + t * (y_3 - y_1))$$

and the bottom right coordinate is defined as:

$$(x_2 + t * (x_4 - x_2), y_2 + t * (y_4 - y_2))$$

Figure 2. Original image (left) and image about 10 feet into the scene (right)

With these definitions, I can use this transition rectangle to move from one image in our field of view stack to a predetermined estimate of the field of view of the next image, and by controlling the increment for t, I can change how quickly I move between two scenes. One issue that arises with cropping images in this fashion is that the number of pixels in our image changes as a function of t and the two field of view rectangles. In order to produce a final video, I need each frame to have the same number of pixels, so for each transition rectangle, I used interp2d to sample pixels inside the transition rectangle such that the size of each image remains the same. Even though each transition image will have the same number of pixels, I would expect the quality of each image to be dependent on the size of the transition rectangle.

2.1. Image Alignment

With this sudo-zoom function defined, I now need to determine the second field of view image. I experimented with a couple different approaches, including estimating and using a homography matrix, and estimating depth using a known object.

2.2. Finding the Homography Matrix

My first thought was to estimate the homography matrix that described the perspective transform between the two consecutive images in the field of view stack. To do so, I used the SIFT technique to extract the features of the two images, and FLANN based matching to match the features between two images which then allows us to find the homography matrix.

I experimented with many parameters and methods to find a homography that worked well consistently. I found that often times, the recovered homography matrix would work well for a portion of the image, but very poorly for other portions. Most of the time, this homography only really worked well on the middle of the image, so in my final implementation, I masked out the middle of the images to try and prevent feature mapping in this portion of the images. Through further experimentation, I found that an 1000 pixel square worked quite well, and this method consistently found a more accurate homography when compared to no masking. An example of this masking is shown

Figure 3. Example masking of image

Figure 4. Estimated Field of View of Second Image

in figure 3, and the result of the estimated field of view is shown in figure 4.

2.3. Smoothing the result

With a homography matrix found, I now want to combine the two separate frames in a way that is not jolting. Due to the separation between the two frames, if I transition between the final cropped image of a scene, and the first cropped image of the very next scene, then this change can often look very strange. For this reason, I experimented with some methods to smoothen the transition, namely, image blending. This is done to mimic the video zoom using a series of photos. By using OpenCV's image blending functions, I can define a new image $g(x)$ such that

$$g(x) = \alpha * f_0(x) + (1 - \alpha) * f_1(x)$$

where $f_0(x)$ is the final cropped image of at a location, and $f_1(x)$ is the very first image at the next location. Figure 5 demonstrates an example of this, where a still shot of each frame is not very compelling, but combining each frame makes transitions smoother and more compelling. Another trick that I found to be helpful was to vary alpha as I transition between the scenes. If I start alpha at 0, and increase it until it reaches 1, then I can transition between scenes better.

Figure 5. Final cropped image of a scene (left), the very first image of the next scene (middle), and our merged result (right)

In order to make merging more accurate, I also aligned the secondary images of the scene with the unfinished transition of the first scene. I found this made the result slightly better, but did not have a huge improvement on quality. Figure 7 demonstrates an example of the wide angle and medium angle lens on a scene, and the estimated field of view of the medium angle image on the wide angle image. I found that this extra information is useful, but it was difficult to sync the images such that the final result had a constant change of depth.

2.4. Other Various Experiments

Much of this project involved coming up with an idea, experimenting with some implementation of it, and trying something else if it didn't work. Because of this, I experimented with many techniques to try and replicate some effect I had seen in other videos. One such effect I had seen frequently is that depth into the scene usually does not scale linearly, as this effect is often used to zoom into a scene far away, where the depth into the scene scales exponentially with time. I experimented with this a bit, and found that it hid the imperfections of the result a little bit, which did make the result more compelling, but this is akin to just making the video faster in order to prevent people from looking too much into the scene. I think if the video I had made zoomed into a very distant point instead of the end of a street (about 100 feet in length) then this technique would work better.

The final experiment I did involved the use of a wide angle lens at location where I took an image with the default lens I had used for the other images (which was a medium angle lens). The use of a wide angle lens allows us to get closer to the setup that established filmmakers have used. I found that the wide angle lens did help in mimicing the zoom effect, but made transitions between locations much more jolting.

3. Results

The final results that I produced were certainly not as compelling as results produced by the filmmakers of Limitless or film-making Youtube channels, which did have more time and resources available to them. The makers of Limitless also took a slightly different approach when creating their effect, where they took the three cameras with a wide angle, medium angle, and telephoto lens, and mounted them as closely as possible, and then shot a video instead of taking multiple photos. This allowed them to spin the camera and make a visually interesting opening scene. This method produces very compelling results, but has the downside of needing three lens, three cameras, some hardware to hold three cameras, and you still have to apply image alignment due to factor for the offset of each camera.

Figure 6. A scene where the sign represents the location of the camera in the next shot

I still found the process of creating this algorithm and the end results to be somewhat compelling, dispite their shortcomings. If this work were to be improved upon, I think some form of depth estimation where the algorithm has access to how much the camera moved could help. In figure 6, I shown an example of some work that I tried with this idea, where I used a large sign with known height and width to represent where the camera would be placed in the next shot. The motivation behind this was that you could use the sign to find an estimate for what the field of view of the next shot would be, and create smoother transitions, but I could not get this method to work better than feature matching between the two images.

The use of one camera and three lenses arose as a more

Figure 7. Wide angle image (left), medium angle image (middle), and estimated field of view of medium angle lens on wide angle image (right)

accessible alternative to those wanted to create this effect. Most of my testing is done on one lens, which creates less compelling results. In the last week, I was able to access a wide-angle lens and a tele-photo lens, but the tele-photo lens had focal ranges that were too large to be useful.

3.1. Results on Wide Angle Lenses

I came across some difficulties with use of the wide angle lens, mainly that it become difficult to linearly move through a scenes, and found that jumps between images become much more apparent. I tried to mitigate this in a few ways but these results were still less compelling to me. I believe with more work on this topic, this approach would work better, especially if a tele lens with a proper focal length.

References

Burger, N. (2011). Limitless. Production Company. Virgin Produced, Rogue, Many Rivers Productions, Boy of the Year, Intermedia Film.

Cinecom.net. (2019). SUPER EASY Infinite Zoom - AFTER EFFECTS (Limitless). Cinecom. https://www.youtube.com/watch?v=1P-SgxQYke4

Lowe, David G. (2004). ”Distinctive image features from scale-invariant keypoints.” Int. J. Comput. Vision, 60(2):91–110,

Lowe, David G. (1999). ”Object recognition from local scale-invariant features” (PDF). Proceedings of the International Conference on Computer Vision. Vol. 2 doi:10.1109/ICCV.1999.790410.