

ECE 418, PROJECT PROPOSAL

DIGITAL ZOOM

Justin Sconza
sconza@illinois.edu

April 12, 2018

PROJECT DESCRIPTION

For my project I will implement a program that performs a digital zoom on an image. To run the program, the user will enter an image file name, a pixel location to zoom in upon and the amount of frames for the digital zoom to take. The output of the program will be a movie file of the digital zoom. As an example, let's say the user wanted to zoom in by a factor of 22, across 10 frames, on the pixel location (310, 418). The terminal input would be as follows.

```
./digitalZoom input.png 22 10 310 418
```

To perform a digital zoom, I will need to repeatedly upsample then downsample an image in order to achieve non-integer magnification ratios for each frame of the digital zoom movie. This process will also require repeated filtering based on the upsample and downsample factors as discussed in the course notes. For example, if the user specifies a ten-frame zoom with a final magnification of 17 : 1, then each frame will need to be $1.7\times$ more magnified than the previous. This means that each frame, except the very first, will be created from the previous by 1), filtering and interpolating by a factor of 17 and then 2), decimating by a factor of 10 and filtering appropriately. The biggest challenge here will be efficiently implementing the filters.

As far as choosing the pixel location to zoom in on, I plan to first make the movie of magnified images and then crop each successive frame to the original size but centered closer and closer to the destination pixel location. If, for example, the image center is labeled (x_0, y_0) , the desired frame count is 10 and the final location is (x_1, y_1) , then after doing the appropriate upsampling/downsampling, each frame will be centered $\left(\frac{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}}{10}\cos(\theta), \frac{\sqrt{(x_1-x_0)^2+(y_1-y_0)^2}}{10}\sin(\theta) \right)$ further away from the origin and closer to the final location, where $\theta = \tan^{-1} \left(\frac{y_1-y_0}{x_1-x_0} \right)$, is the angle from the image center to the final pixel location.

There will probably also need to be restrictions on where the final pixel location can be. For example, if the user chooses an edge for the final location, it won't be possible to center the last frames. Most likely, the final pixel locations will be restricted to live within a box of $\frac{\text{image size}}{4} \times \frac{\text{image size}}{4}$, drawn around the image center.