Project #5: Image Manipulation in a "Magic Lens"

Haoxiang Wang; Student ID: 932359049

February 18, 2016

This project is the fifth project I have done in this class, and it is the first project that using GLSL to implement image processing effects. Since I gained the experience of image processing from my research project, this project seems easy to deal with, and the result turned out that the project was also fun to play with. The whole project only takes me a few hours to get everything done and fit to all the requirements. The results images and the explanation of how code works will be described in the after section.

1 Source Listing

In this project, three files are necessary. They are .glib file, .vert file, and .frag file. No .obj file is needed in this project. The specific filenames and usages are listed below.

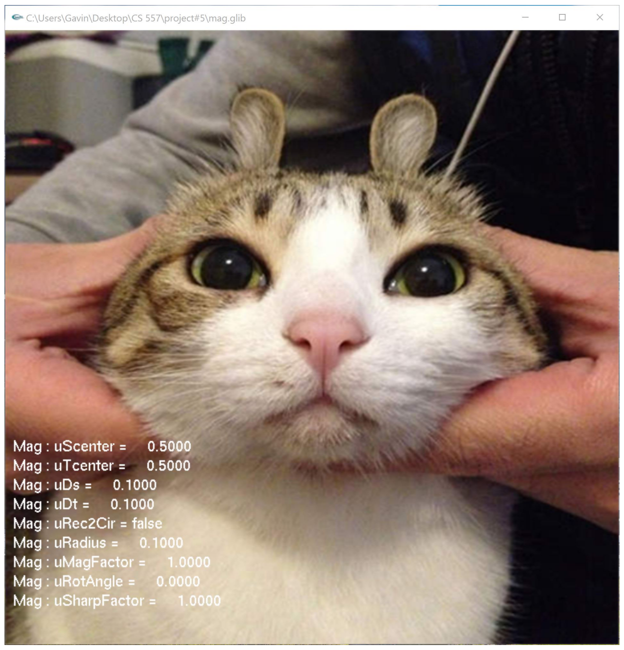
mag.glib — Handle the user interface and object

mag.vert — Handle the vertex shader

mag.frag — handle the fragment shader

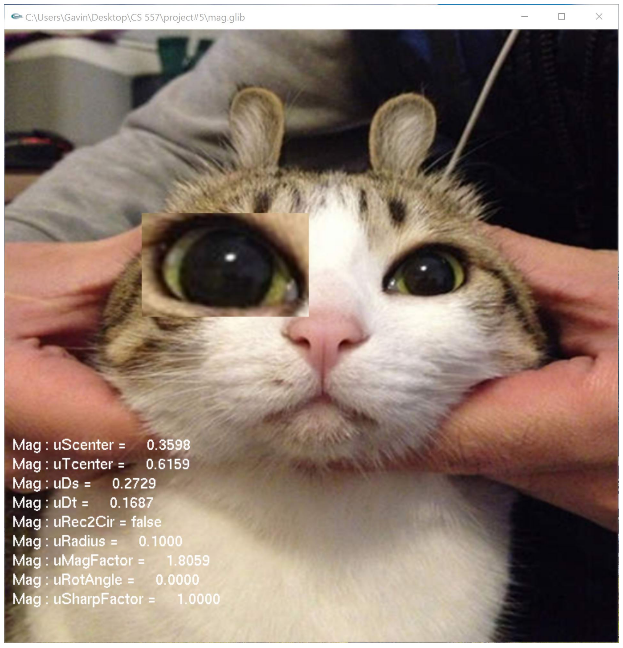
2 Result Images and Explanation

The requirements of this project can be simply divided into three parts and one extra part. The whole project is trying to implement a "magic lens" which contains the magnification, rotation, and sharpening. The first one is to implement the magnification, the second part is to implement the rotation, and the third part is to implement sharpening. The extra part of this project is changing the rectangle lens to a circle lens, which requires some extra work in the code. The image we are trying to add some effects on is treated as a texture mapped onto an X-Y Quad. The following image shows that original image and the variable values when the image is rendered.

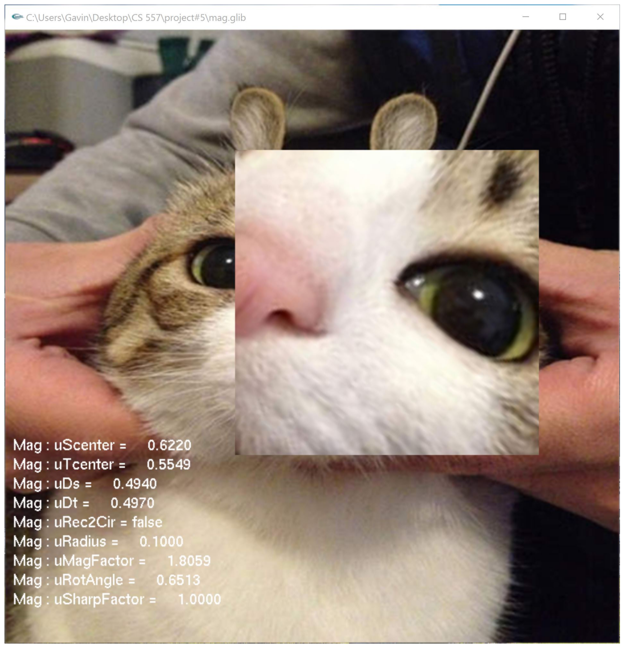

C:\Users\Gavin\Desktop\CS 557\project#5\mag.glib

```
Mag : uScenter =     0.5000
Mag : uTcenter =     0.5000
Mag : uDs =     0.1000
Mag : uDt =     0.1000
Mag : uRec2Cir = false
Mag : uRadius =     0.1000
Mag : uMagFactor =     1.0000
Mag : uRotAngle =     0.0000
Mag : uSharpFactor =     1.0000
```

The first part of the project is magnification. The part of the image covered by the lens will be "enlarged" by a coefficient getting from the slider named "uMagFactor". The effect is not the real magnification, it doesn't even contain the refraction implementation. the fact behind my magnification is playing with the st coordinates. By checking if the texel is in the $[uScenter - uDs/2, uScenter + uDs/2]$ and $[uTcenter - uDt/2, uTcenter + uDt/2]$ range, we could tell whether the texel should be manipulate or not. If the texel needs to be manipulated, I simply grab the rgb color from another texel and put it as itself's rgb value. The equations for searching the texels are $sp = (s - uScenter)/uMagFactor + uScenter$ and $tp = (t - uTcenter)/uMagFactor + uTcenter$. The following left image shows the effect of magnification on cat's eye.


C:\Users\Gavin\Desktop\CS 557\project#5\mag.glib

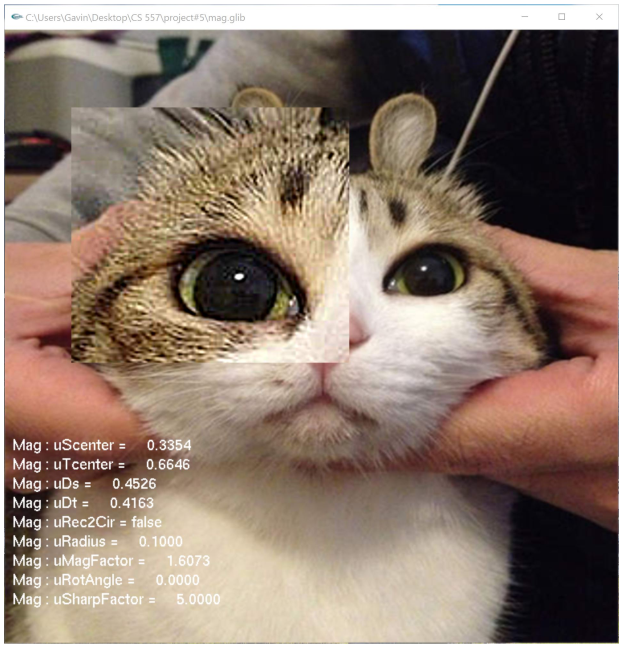
```
Mag : uScenter =    0.3598
Mag : uTcenter =    0.6159
Mag : uDs =    0.2729
Mag : uDt =    0.1687
Mag : uRec2Cir = false
Mag : uRadius =    0.1000
Mag : uMagFactor =    1.8059
Mag : uRotAngle =    0.0000
Mag : uSharpFactor =    1.0000
```


C:\Users\Gavin\Desktop\CS 557\project#5\mag.glib

```
Mag : uScenter =    0.6220
Mag : uTcenter =    0.5549
Mag : uDs =    0.4940
Mag : uDt =    0.4970
Mag : uRec2Cir = false
Mag : uRadius =    0.1000
Mag : uMagFactor =    1.8059
Mag : uRotAngle =    0.6513
Mag : uSharpFactor =    1.0000
```

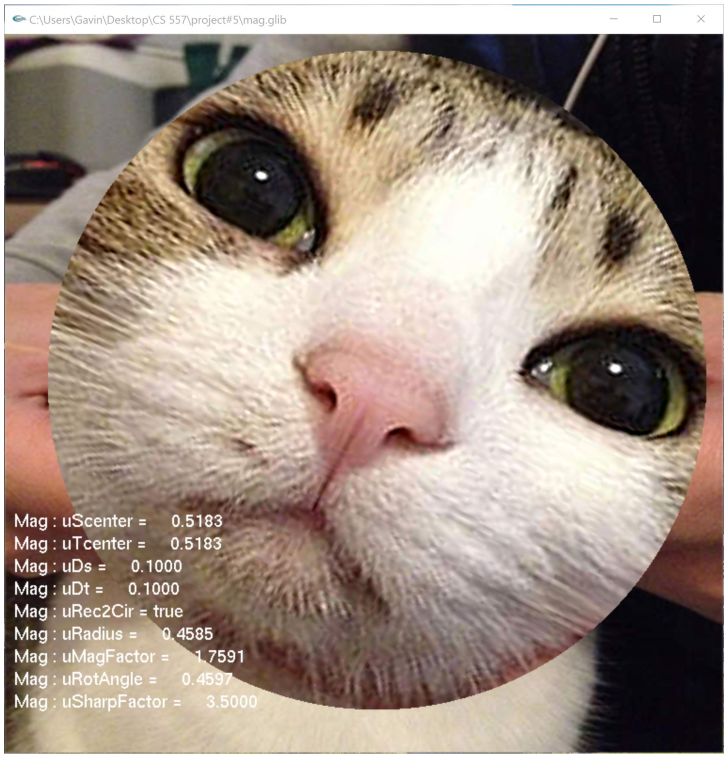
The second part of the project if rotation. The image covered by the lens will be rotate an angle from the slider "uRotAngle". As it mentioned in the project website, to rotate the image we need to apply the equations $x' = x * cos(theta) - y * sin(theta)$ and $y' = x * sin(theta) + y * cos(theta)$. However for this assignment, we need to manipulate the st coordinates instead of the xt coordinates, and the rotate center should be the center of the lens, so I created the following equations instead: $sp = uScenter + ((stp.s - uScenter) * cos(uRotAngle) - (stp.t - uTcenter) * sin(uRotAngle))$ and $tp = uTcenter + ((stp.s - uScenter) * sin(uRotAngle) + (stp.t - uTcenter) * cos(uRotAngle))$. The rotation effect is shown in the above right image.

The third part of this project is the sharpening. To implement the sharpening we simply apply the blur matrix to the image. the blur matrix simply blur the part that we don't need in the image in order to deliver the sharpening effects. The example code has already been given in the class notes, and the changes that need to be applied to it are just changing the original st coordinates to the current coordinates that have been enlarged and rotated. The following image shows the effect of sharpening. The details on the fur of the cat delivered a clear idea of how the sharpening works.



```
Mag : uScenter =     0.3354
Mag : uTcenter =     0.6646
Mag : uDs =     0.4526
Mag : uDt =     0.4163
Mag : uRec2Cir = false
Mag : uRadius =     0.1000
Mag : uMagFactor =     1.6073
Mag : uRotAngle =     0.0000
Mag : uSharpFactor =     5.0000
```

The final part of the project is the extra credit part. It requires changing the lens from rectangle to circle. Basically, what we have done for scaling, rotation and sharpening stays the same, and the only thing that needs to be changed is the way we determine whether the texel is in the lens or not. Instead of checking the boundary of the rectangle, I simply calculate the distance from the selected texel to the lens center. If the distance is smaller than the radius of the circle, then it is in the lens, otherwise it's not. The effects of circle lens with scaling, rotation and sharpening is shown in the following image.



```
Mag : uScenter =     0.5183
Mag : uTcenter =     0.5183
Mag : uDs =     0.1000
Mag : uDt =     0.1000
Mag : uRec2Cir = true
Mag : uRadius =     0.4585
Mag : uMagFactor =     1.7591
Mag : uRotAngle =     0.4597
Mag : uSharpFactor =     3.5000
```

3 Summary

This project is pretty easy to handle, and the effects is really fun to play with. Manipulating the st coordinates gives me a deeper understanding of the texture. I'm looking forward to other projects in this class and eager to do them well.