RenderMan and OpenGL Shaders

CS557

Project # 5

Chao Zhang

1. Source listings

```
##OpenGL GLIB
Ortho  -5 5.  -5. 5.
LookAt 0 0 2  0 0 0  0 1 0

Texture2D 5 image.bmp


Vertex   mag.vert
Fragment mag.frag
Program  Mag                                          \
        Circle          <false>                 \
        MagFactor <.1 1. 25.>                   \
        Scenter <0. .5 1.>                              \
        Tcenter <0. .5 1.>                              \
        Ds      <0.01 .1 .5>                            \
        Dt      <0.01 .1 .5>                            \
        RotAngle <-3.14159 0. 3.14159>          \
        SharpFactor <0. 1. 5.>                  \
        ImageUnit  5

QuadXY .2 5.
```

mag.glib

This is the glib file. This file includes all the basic information about this project. The shader type which is vertex shader and the fragment shader. The circle's default is false, so the Magic Lens is a rectangle. If change the circle to true, the Magic Lens will be a circle. The Ds and Dt will change the size of the Lens. The Scenter and Tcenter changes the position of the Lens. MagFactor, Rotangle and sharpFactor will change the magnified, rotated, and sharpened.

```
#version 330 compatibility

out vec2  vST;
void
main( )
{
        vST = gl_MultiTexCoord0.st;
        gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
}
```

mag.vert

This project almost have nothing to do with the vertex sharder. Just the basic coordinate information.

```
#version 330 compatibility

uniform bool            Circle;
uniform float           Scenter;
uniform float           Tcenter;
uniform float           Ds;
uniform float           Dt;
uniform float           MagFactor;
uniform float           RotAngle;
uniform float       SharpFactor;
uniform sampler2D   ImageUnit;


in vec2  vST;
float    ResS, ResT;


void
main( )
{
        float s = vST.s;
        float t = vST.t;
        vec3 color  = texture2D( ImageUnit, vST ).rgb;

        ivec2 ires = textureSize( ImageUnit, 0 );
        ResS = float( ires.s );
        ResT = float( ires.t );


        vec2 st = vST;


        float top = Scenter + Ds;
        float bottom = Scenter - Ds;
        float right = Tcenter + Dt;
        float left = Tcenter - Dt;
```

part 1 mag.frag

This is the first part of the fragment shader.

ivec2 ires = textureSize(ImageUnit, 0);

ResS = float(ires.s);

ResT = float(ires.t); Those lines treat the image as a texture. The lines under that gives the range of the lens.

```
if (Circle)
{
        if((s - Scenter)*(s - Scenter) + (t - Tcenter)*(t - Tcenter) < (Ds) * (Ds) )
        {
                s = s - Scenter;
                t = t - Tcenter;
                s = s * 1.0 / MagFactor;
                t = t * 1.0 / MagFactor;
                float X = s*cos(RotAngle) - t*sin(RotAngle) + Scenter;
                float Y = s*sin(RotAngle) + t*cos(RotAngle) + Tcenter;

                vec2 m = vec2(X,Y);
                vec3 n = texture2D(ImageUnit, m).rgb;

                vec2 stp0 = vec2(1./ResS,  0. );
                vec2 st0p = vec2(0.     ,  1./ResT);
                vec2 stpp = vec2(1./ResS,  1./ResT);
                vec2 stpm = vec2(1./ResS, -1./ResT);
                vec3 i00 =   texture2D( ImageUnit, m ).rgb;
                vec3 im1m1 = texture2D( ImageUnit, m-stpp ).rgb;
                vec3 ip1p1 = texture2D( ImageUnit, m+stpp ).rgb;
                vec3 im1p1 = texture2D( ImageUnit, m-stpm ).rgb;
                vec3 ip1m1 = texture2D( ImageUnit, m+stpm ).rgb;
                vec3 im10 =  texture2D( ImageUnit, m-stp0 ).rgb;
                vec3 ip10 =  texture2D( ImageUnit, m+stp0 ).rgb;
                vec3 i0m1 =  texture2D( ImageUnit, m-st0p ).rgb;
                vec3 i0p1 =  texture2D( ImageUnit, m+st0p ).rgb;
                vec3 target = vec3(0.,0.,0.);
                target += 1.*(im1m1+ip1m1+ip1p1+im1p1);
                target += 2.*(im10+ip10+i0m1+i0p1);
                target += 4.*(i00);
                target /= 16.;
                gl_FragColor = vec4( mix( target, n, SharpFactor ), 1. );
        }
```

part 2 mag.frag

This is the part 2 of the fragment shader. This part is for the circle lens. I use the formula to find if it is inside the circle lens or not. If inside the lens,

s = s - Scenter;

t = t - Tcenter;

s = s * 1.0 / MagFactor;

t = t * 1.0 / MagFactor; this can do the magnify on the inside part. It use the old length multiple the 1/MagFactor to get the new length.

float X = s*cos(RotAngle) - t*sin(RotAngle) + Scenter;

float Y = s*sin(RotAngle) + t*cos(RotAngle) + Tcenter;

Those are the rotation formula. The reason why it need to plus the Scenter or Tcenter is the inside part need to rotation around the center. The sharpening function is come from the class formula. Because it is on the vertex after no matter rotation or magnify, so I create the m as the new position. For the vertex outside the lens, it didn't change.

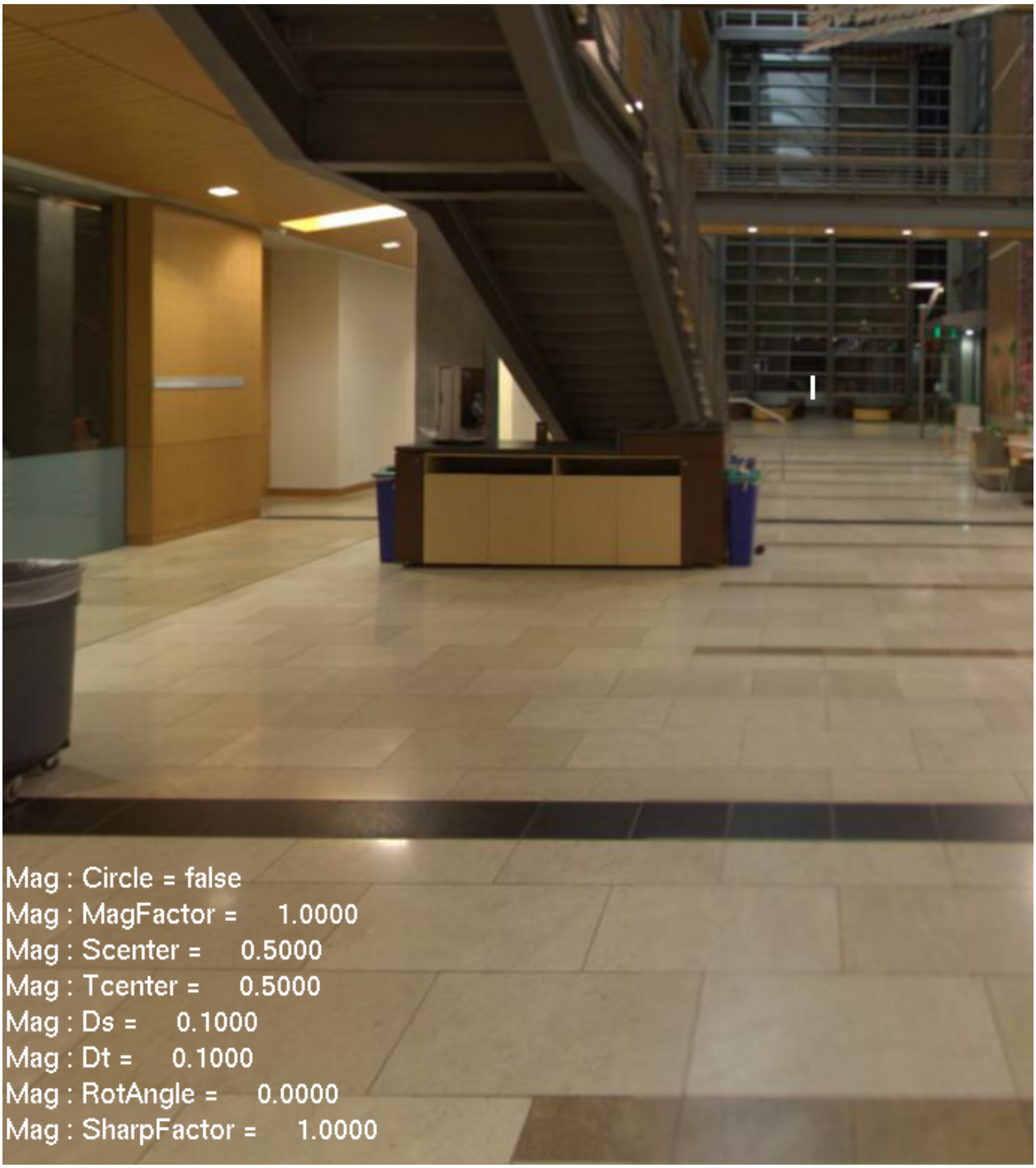
```
} else{
        float top = Scenter + Ds;
        float bottom = Scenter - Ds;
        float right = Tcenter + Dt;
        float left = Tcenter - Dt;
        if( s < top && s > bottom && t > left && t < right )
        {
                s = s - Scenter;
                t = t - Tcenter;
                s = s * 1.0 / MagFactor;
                t = t * 1.0 / MagFactor;
                float X = s*cos(RotAngle) - t*sin(RotAngle) + Scenter;
                float Y = s*sin(RotAngle) + t*cos(RotAngle) + Tcenter;
                vec2 m = vec2(X,Y);
                vec3 n = texture2D(ImageUnit, m).rgb;

                vec2 stp0 = vec2(1./ResS,  0. );
                vec2 st0p = vec2(0.     ,  1./ResT);
                vec2 stpp = vec2(1./ResS,  1./ResT);
                vec2 stpm = vec2(1./ResS, -1./ResT);
                vec3 i00 =   texture2D( ImageUnit, m ).rgb;
                vec3 im1m1 = texture2D( ImageUnit, m-stpp ).rgb;
                vec3 ip1p1 = texture2D( ImageUnit, m+stpp ).rgb;
                vec3 im1p1 = texture2D( ImageUnit, m-stpm ).rgb;
                vec3 ip1m1 = texture2D( ImageUnit, m+stpm ).rgb;
                vec3 im10 =  texture2D( ImageUnit, m-stp0 ).rgb;
                vec3 ip10 =  texture2D( ImageUnit, m+stp0 ).rgb;
                vec3 i0m1 =  texture2D( ImageUnit, m-st0p ).rgb;
                vec3 i0p1 =  texture2D( ImageUnit, m+st0p ).rgb;
                vec3 target = vec3(0.,0.,0.);
                target += 1.*(im1m1+ip1m1+ip1p1+im1p1);
                target += 2.*(im10+ip10+i0m1+i0p1);
                target += 4.*(i00);
                target /= 16.;
                gl_FragColor = vec4( mix( target, n, SharpFactor ), 1. );
        }
        else
        {
                gl_FragColor = vec4( color, 1. );
        }
}
```

part 3 mag.frag

This part is the same as the part 2, the only different is the lens is a rectangle. So I use the different formula to check if it is inside the lens or not.

2. Result

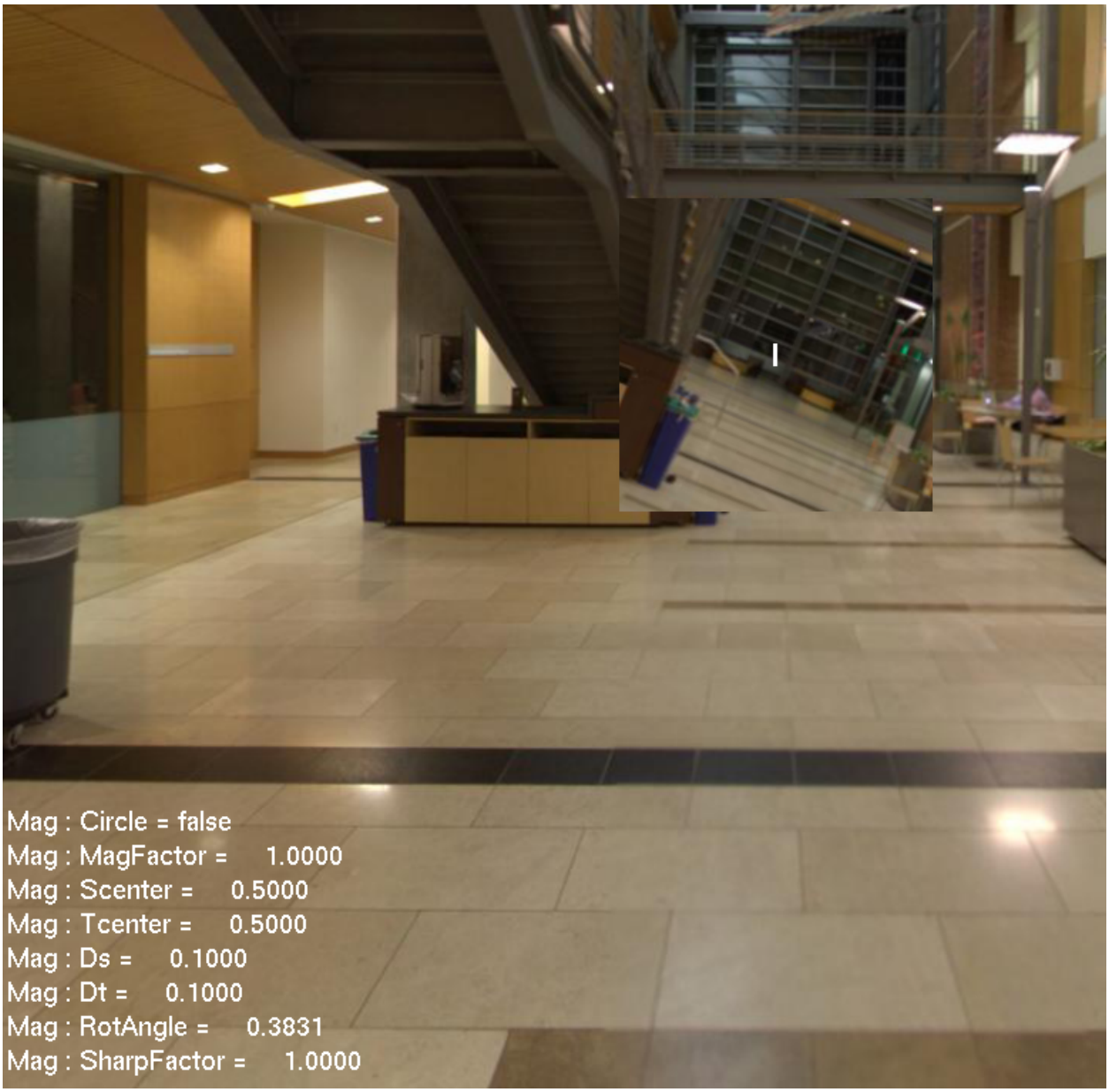

Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.5000
Mag : Tcenter = 0.5000
Mag : Ds = 0.1000
Mag : Dt = 0.1000
Mag : RotAngle = 0.0000
Mag : SharpFactor = 1.0000


Mag : Circle = false
Mag : MagFactor = 1.6073
Mag : Scenter = 0.3780
Mag : Tcenter = 0.4207
Mag : Ds = 0.1000
Mag : Dt = 0.1000
Mag : RotAngle = 0.0000
Mag : SharpFactor = 1.0000


Mag : Circle = false
Mag : MagFactor = 0.5445
Mag : Scenter = 0.3780
Mag : Tcenter = 0.4207
Mag : Ds = 0.1000
Mag : Dt = 0.1000
Mag : RotAngle = 0.0000
Mag : SharpFactor = 1.0000


Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.5000
Mag : Tcenter = 0.5000
Mag : Ds = 0.1721
Mag : Dt = 0.1576
Mag : RotAngle = 0.3831
Mag : SharpFactor = 1.0000


Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.5000
Mag : Tcenter = 0.5000
Mag : Ds = 0.0764
Mag : Dt = 0.0620
Mag : RotAngle = 0.3831
Mag : SharpFactor = 1.0000


Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.5000
Mag : Tcenter = 0.5000
Mag : Ds = 0.1000
Mag : Dt = 0.1000
Mag : RotAngle = 0.3831
Mag : SharpFactor = 1.0000


Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.3293
Mag : Tcenter = 0.2378
Mag : Ds = 0.0585
Mag : Dt = 0.1008
Mag : RotAngle = 0.3831
Mag : SharpFactor = 1.0000

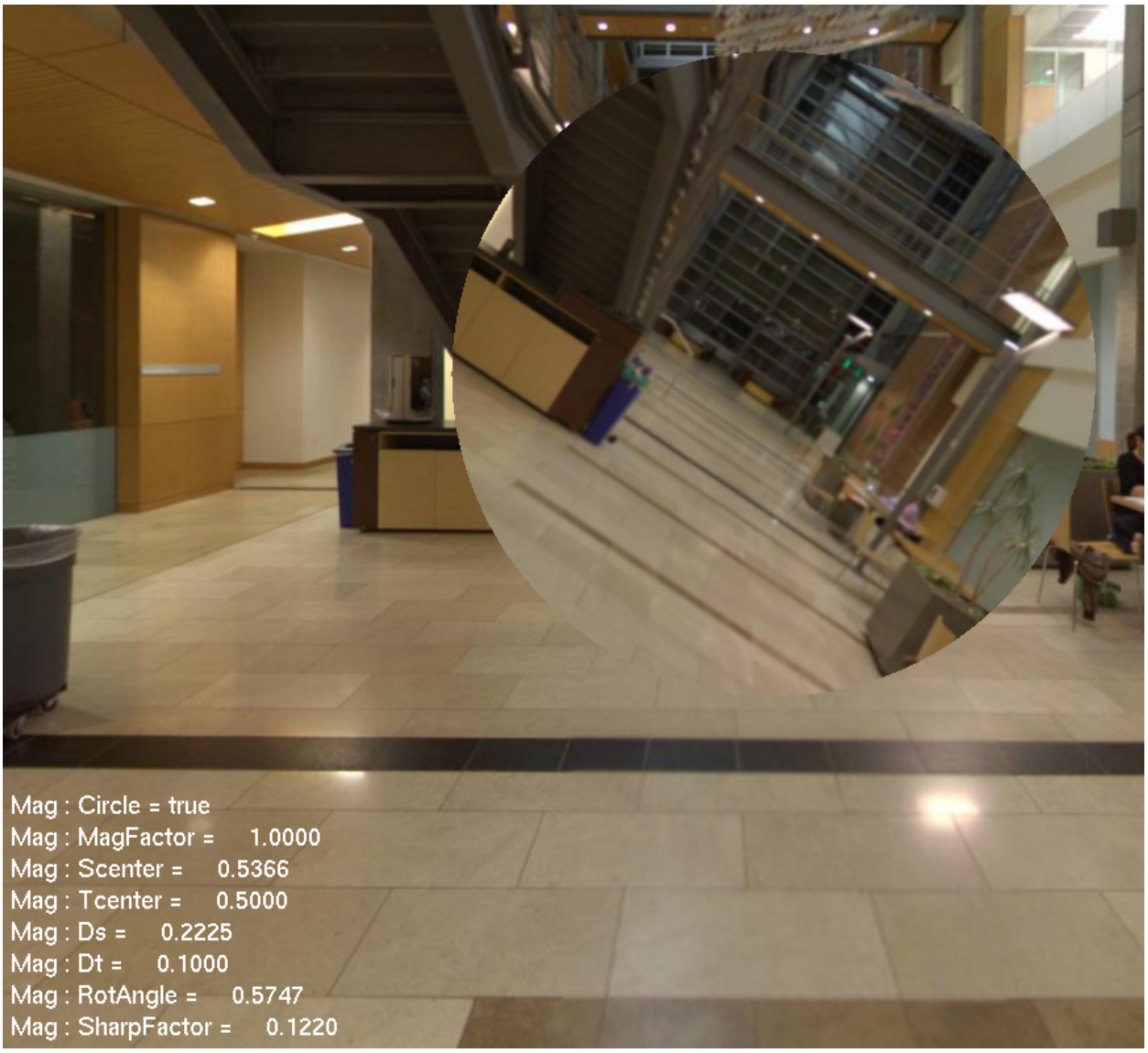

Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.3232
Mag : Tcenter = 0.2561
Mag : Ds = 0.1000
Mag : Dt = 0.1000
Mag : RotAngle = 0.5747
Mag : SharpFactor = 0.0000


Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.3232
Mag : Tcenter = 0.2561
Mag : Ds = 0.1000
Mag : Dt = 0.1000
Mag : RotAngle = 0.5747
Mag : SharpFactor = 1.0016


Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.3232
Mag : Tcenter = 0.2561
Mag : Ds = 0.1000
Mag : Dt = 0.1000
Mag : RotAngle = 0.5747
Mag : SharpFactor = 5.0000


Mag : Circle = false
Mag : MagFactor = 1.0000
Mag : Scenter = 0.3293
Mag : Tcenter = 0.2378
Mag : Ds = 0.1123
Mag : Dt = 0.0620
Mag : RotAngle = 0.8046
Mag : SharpFactor = 1.0000


Mag : Circle = true
Mag : MagFactor = 1.0000
Mag : Scenter = 0.3232
Mag : Tcenter = 0.2561
Mag : Ds = 0.1000
Mag : Dt = 0.1000
Mag : RotAngle = 0.5747
Mag : SharpFactor = 5.0000


Mag : Circle = true
Mag : MagFactor = 1.0000
Mag : Scenter = 0.5366
Mag : Tcenter = 0.5000
Mag : Ds = 0.2225
Mag : Dt = 0.1000
Mag : RotAngle = 0.5747
Mag : SharpFactor = 0.1220