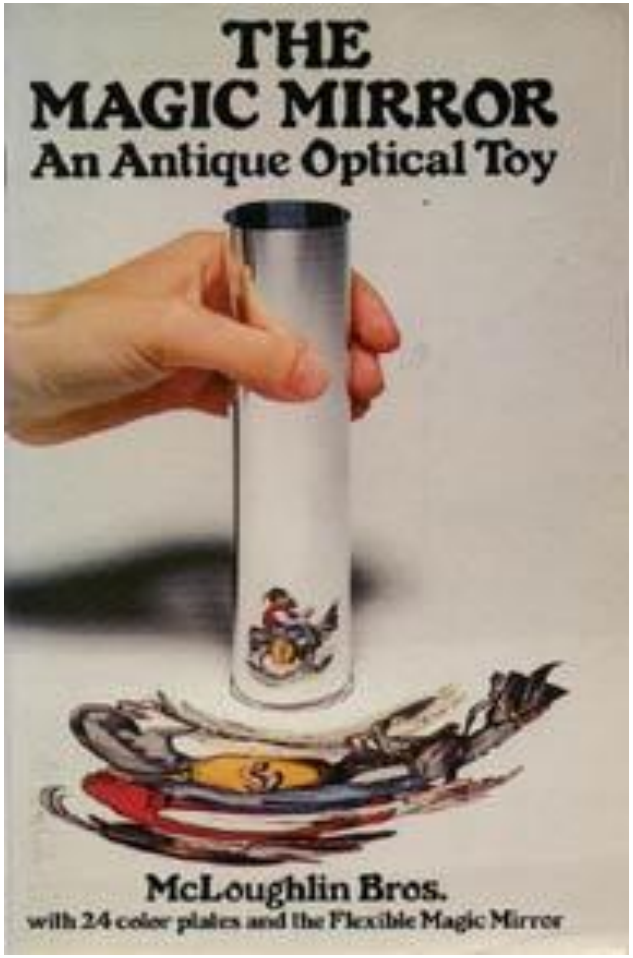
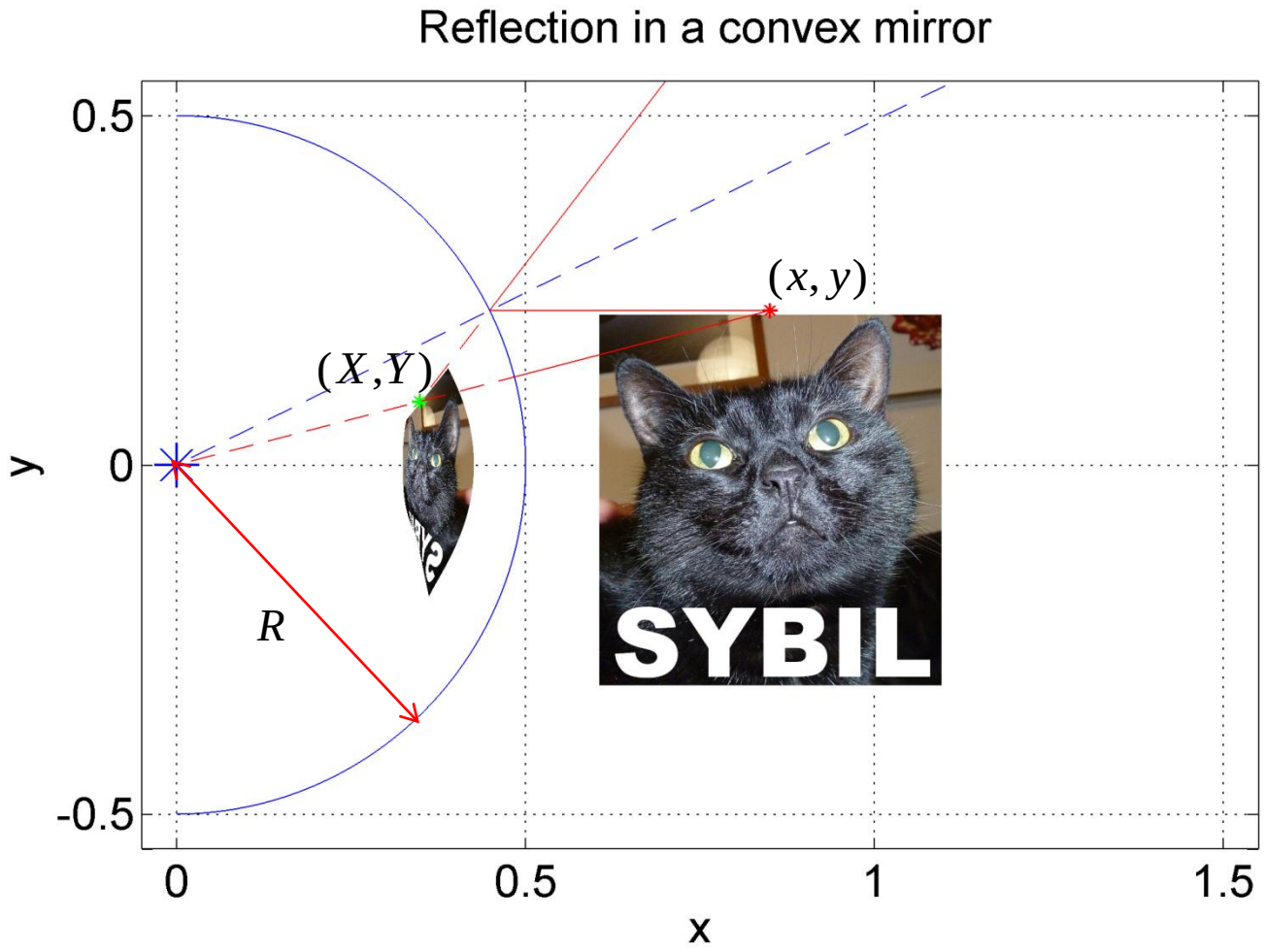

Andy French Dec 2021
Anamorphic
images


An anamorphosis painting of a perspective stern view of a ship model.
Artist unknown, c.1744-1774. Oil on panel.
Science Museum Mathematics Gallery 2021



Sancho Panza on his donkey. The fictional squire to *Don Quixote* in Cervantes' 1605 novel.

[Anamorphic art]



$$\alpha = \tfrac{1}{2}\tan^{-1}\left(\tfrac{y}{x}\right)$$

$$k = \frac{x}{\cos(2\alpha)}$$

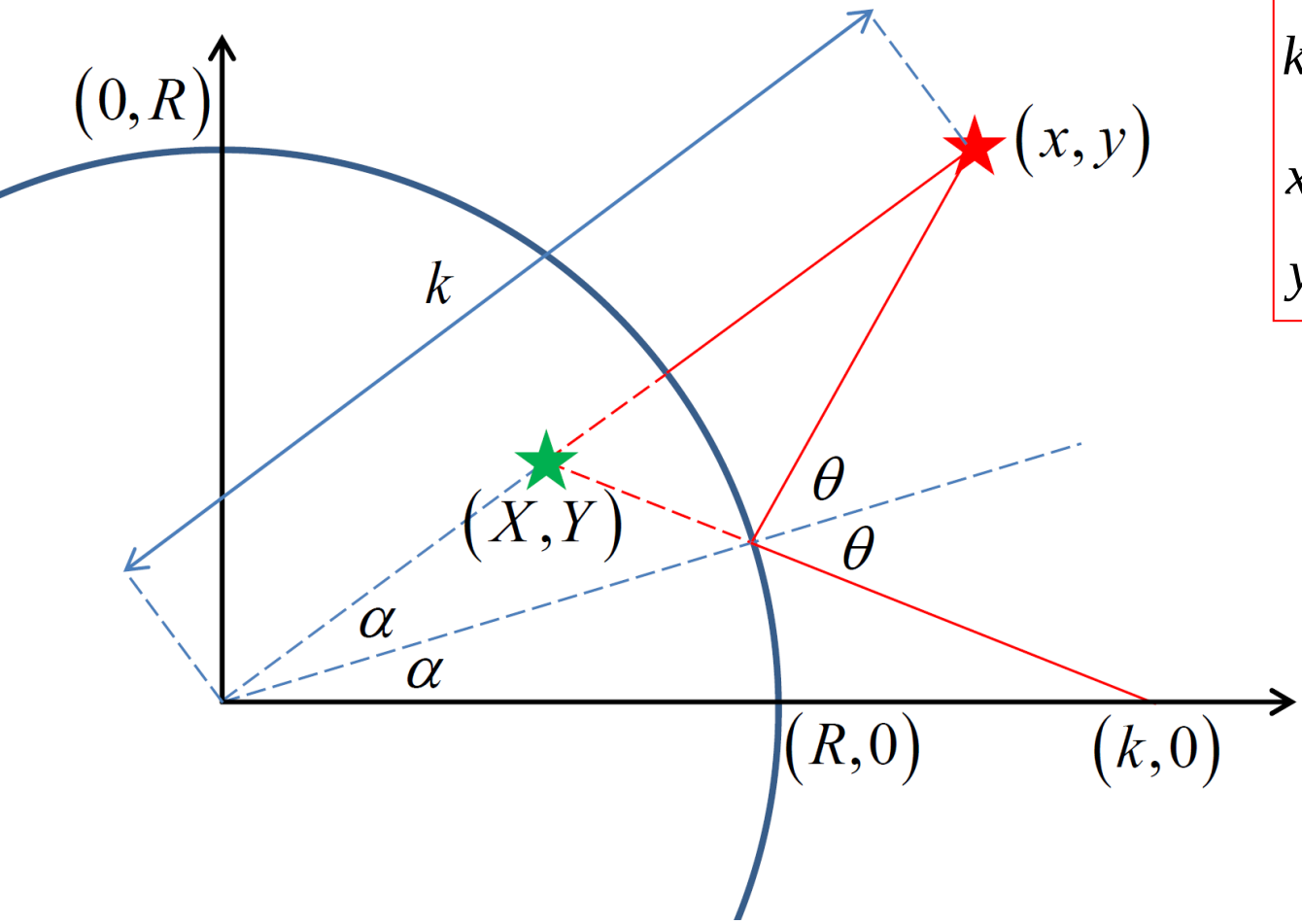
$$Y = \frac{k\sin\alpha}{\frac{k}{R} - \cos\alpha + \frac{x}{y}\sin\alpha}$$

$$X = x\frac{Y}{y}$$

Virtual image from object coordinates

We see an upright, distorted *virtual image* in a cylindrical mirror.

i.e. the *apparent source* of (diverging) light rays from the mirror



$$\alpha = \tfrac{1}{2}\tan^{-1}\left(\tfrac{Y}{X}\right)$$

$$k = \frac{R\left(Y\cos\alpha - X\sin\alpha\right)}{Y - R\sin\alpha}$$

$$x = k\cos\left(2\alpha\right)$$

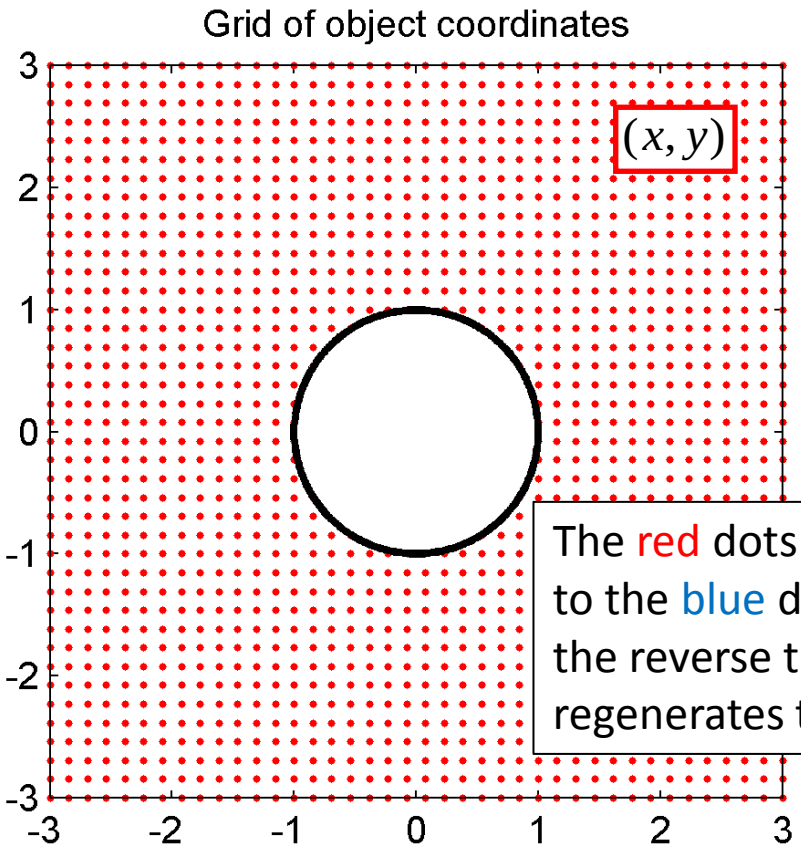
$$y = k\sin\left(2\alpha\right)$$

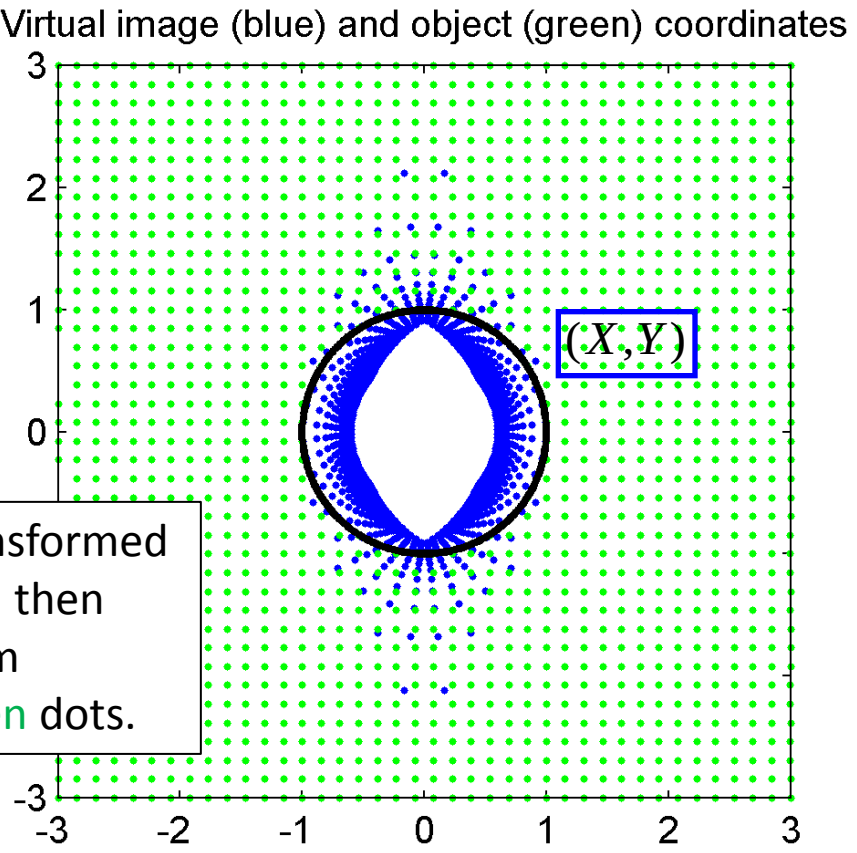
$$\alpha = \tfrac{1}{2}\tan^{-1}\left(\tfrac{y}{x}\right)$$

$$k = \frac{x}{\cos\left(2\alpha\right)}$$

$$Y = \frac{k\sin\alpha}{\frac{k}{R} - \cos\alpha + \frac{x}{y}\sin\alpha}$$

$$X = x\frac{Y}{y}$$





The red dots are transformed to the blue dots, and then the reverse transform regenerates the green dots.

$$\alpha = \tfrac{1}{2}\tan^{-1}\left(\tfrac{y}{x}\right)$$

$$k = \frac{x}{\cos(2\alpha)}$$

$$Y = \frac{k\sin\alpha}{\frac{k}{R} - \cos\alpha + \frac{x}{y}\sin\alpha}$$

$$X = x\frac{Y}{y}$$

Virtual image from object coordinates

$$\alpha = \tfrac{1}{2}\tan^{-1}\left(\tfrac{Y}{X}\right)$$

$$k = \frac{R\left(Y\cos\alpha - X\sin\alpha\right)}{Y - R\sin\alpha}$$

$$x = k\cos(2\alpha)$$

$$y = k\sin(2\alpha)$$

Object coordinates from virtual image coordinates

KISS OF CHYTRID
Jonty Hurwitz 2009-2010

https://jontyhurwitz.com/

https://twistedsifter.com/2013/04/anamorphic-sculptures-made-with-algorithms-jonty-hurwitz/

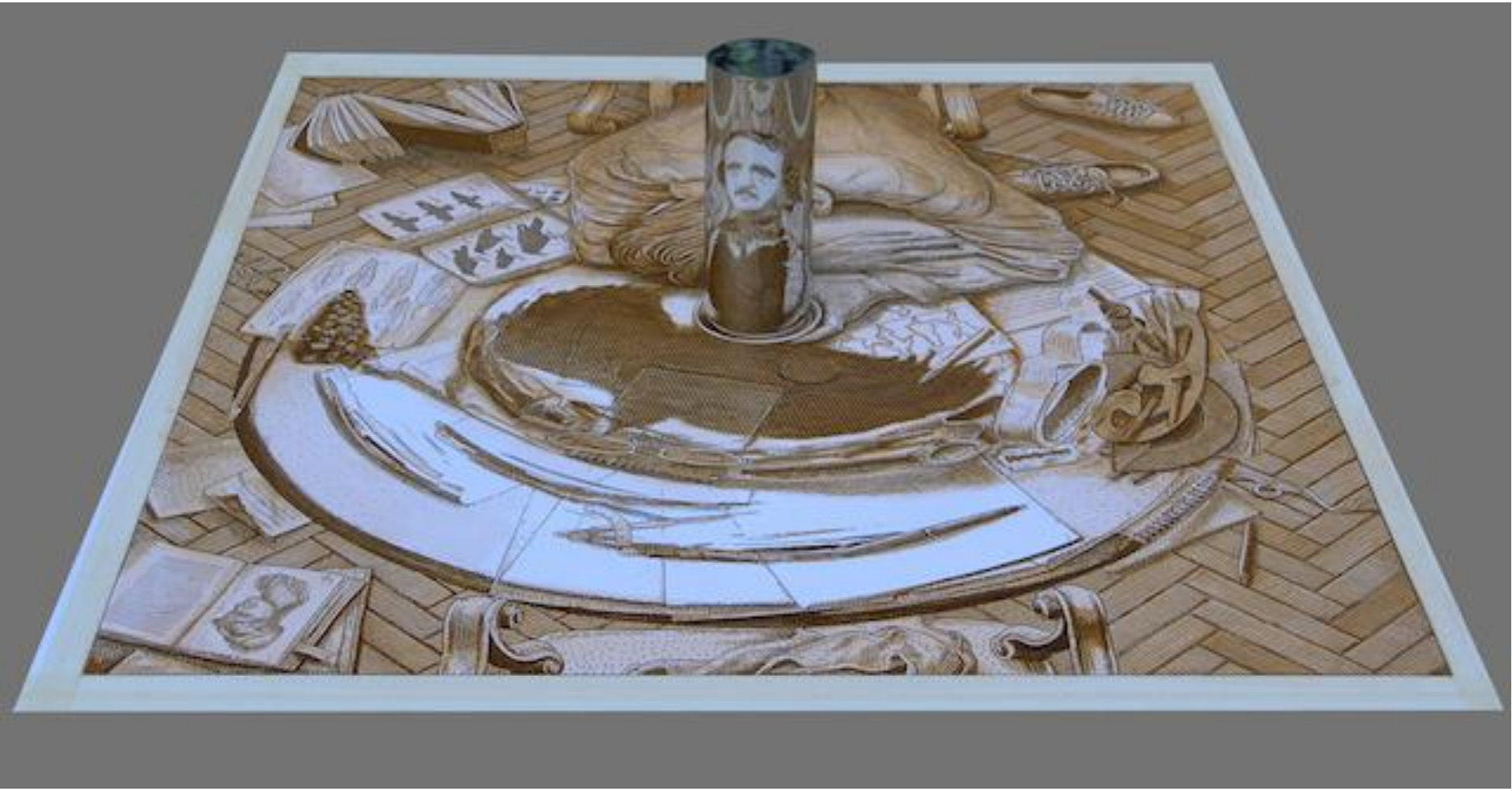
REJUVENATION
Jonty Hurwitz 2008

YOGI BANKER 2010

István Orosz

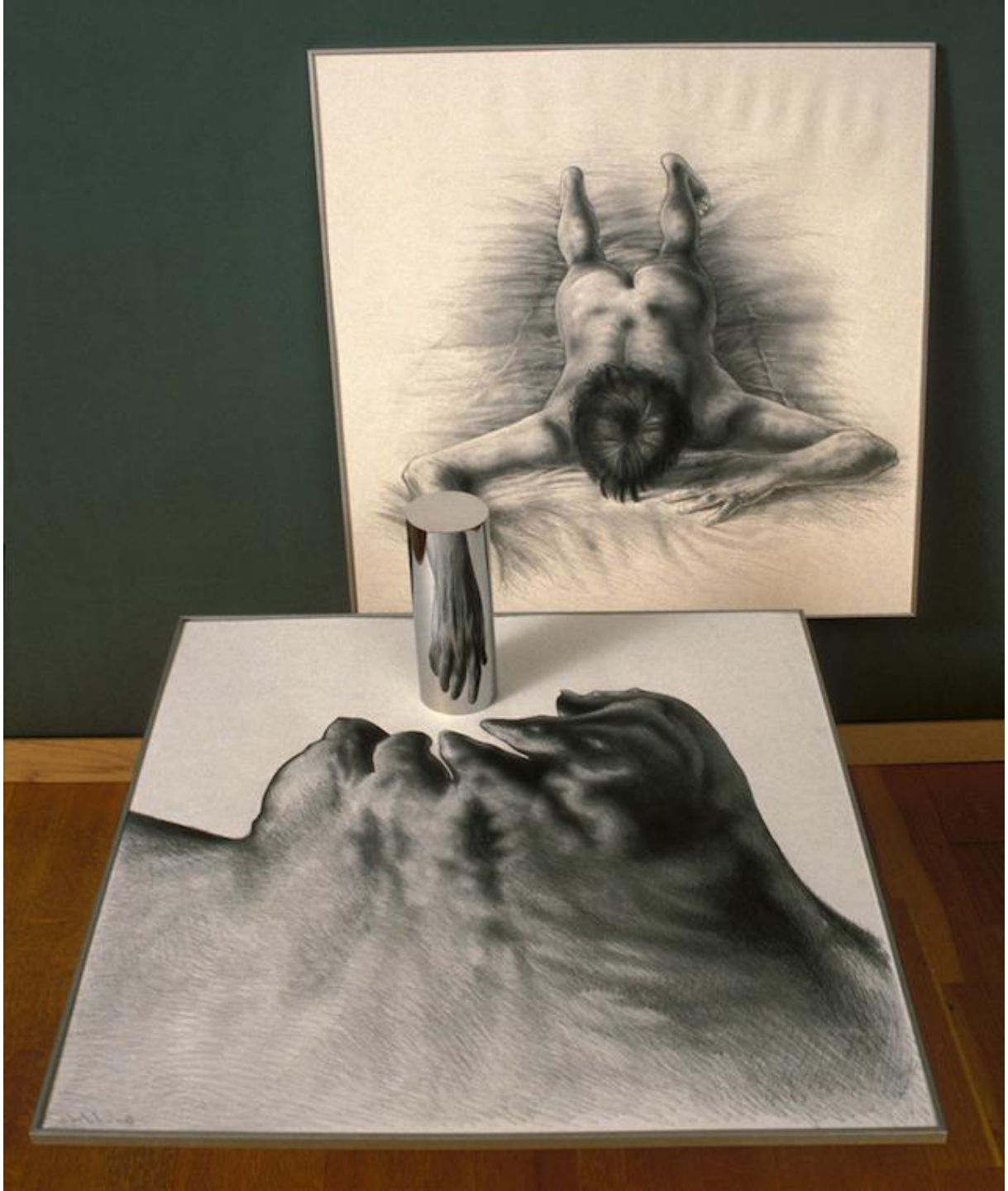
http://www.fubiz.net/en/2015/02/10/anamorphic-work-with-cylindrical-mirror-2/

István Orosz



István Orosz

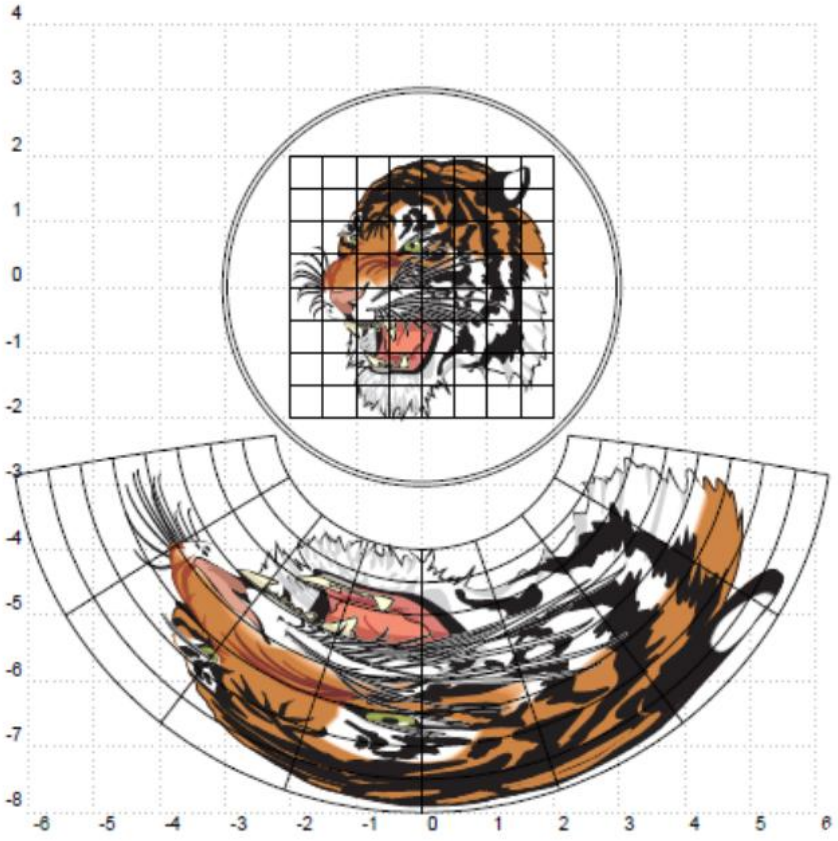
István Orosz



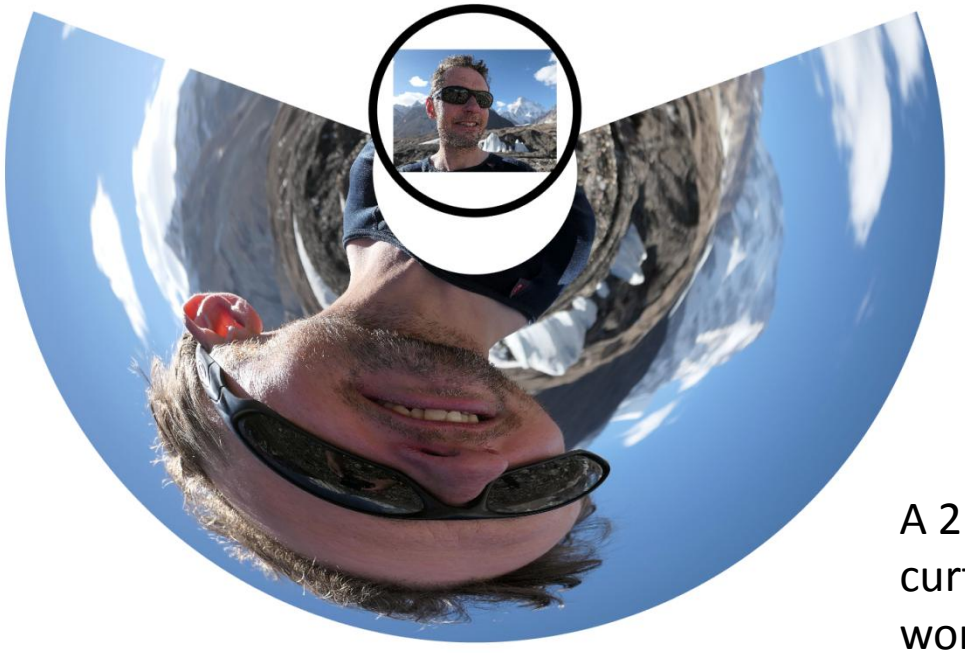
István Orosz

I originally attempted a transformation based upon the convex mirror equations. However – I ran into several problems! Do you need a three-dimensional virtual image? If so, what should I do with a 2D photo? (This is really the ideal input). What about the viewing position. How do heights transform?

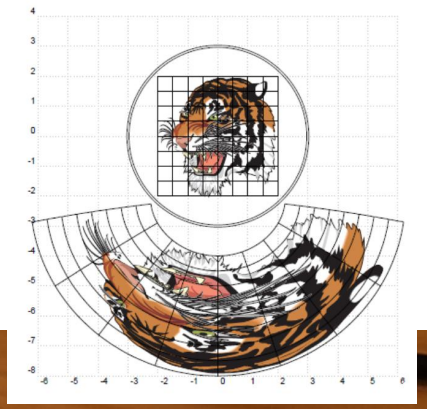
Eventually I spotted the following illustration, which suggest a much simpler 2D Cartesians to plane polar transformation. It is certainly not an exact virtual to object coordinate mapping, but offers something that is both simple, and appears to work well enough.



This Instructable was also useful.

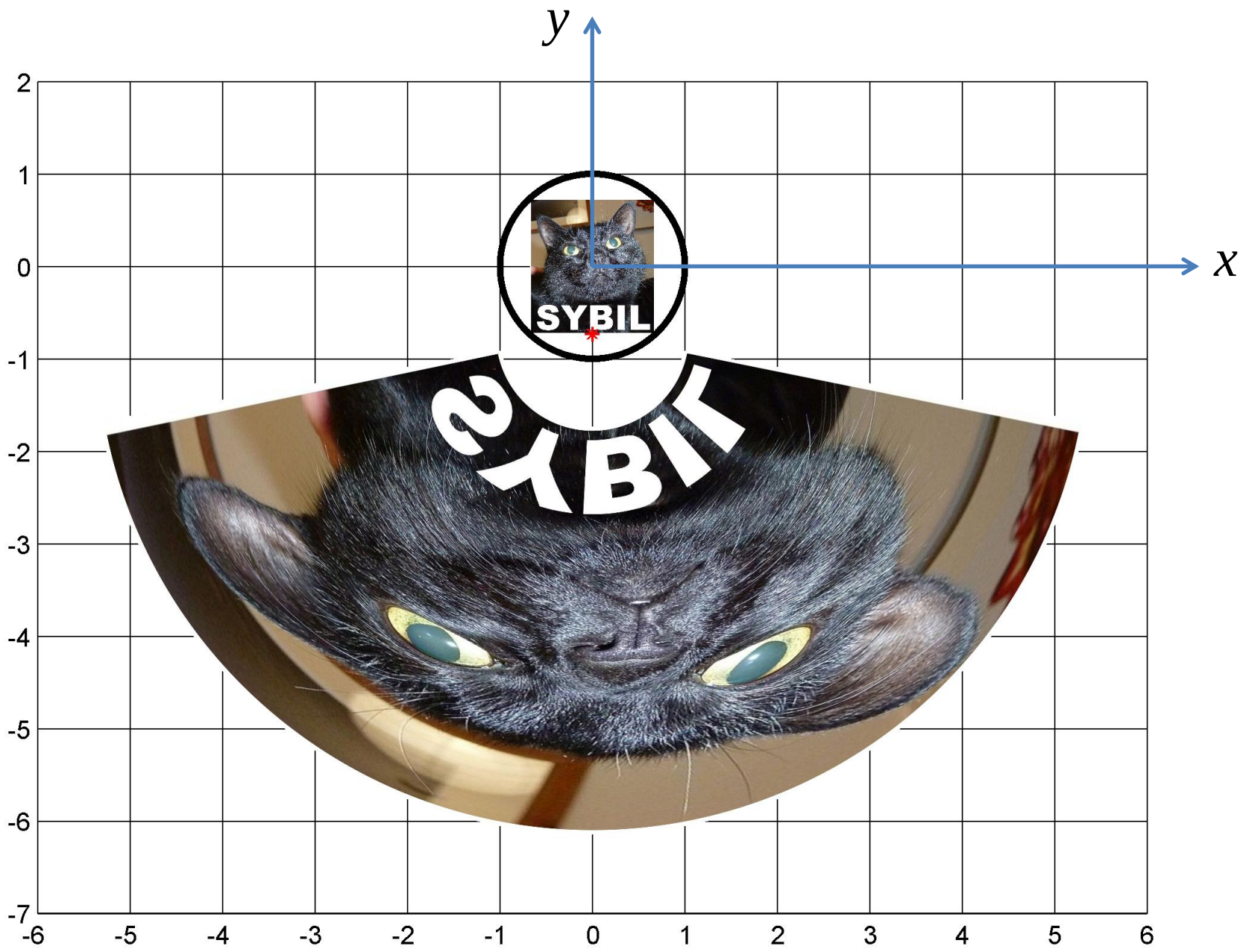


A MATLAB program `anamorph.m` fits *any* bitmap image into a unit circle, and then calculates an anamorphic projection based upon a mapping of a rectangular grid inside the black circle to a circle sector beyond.



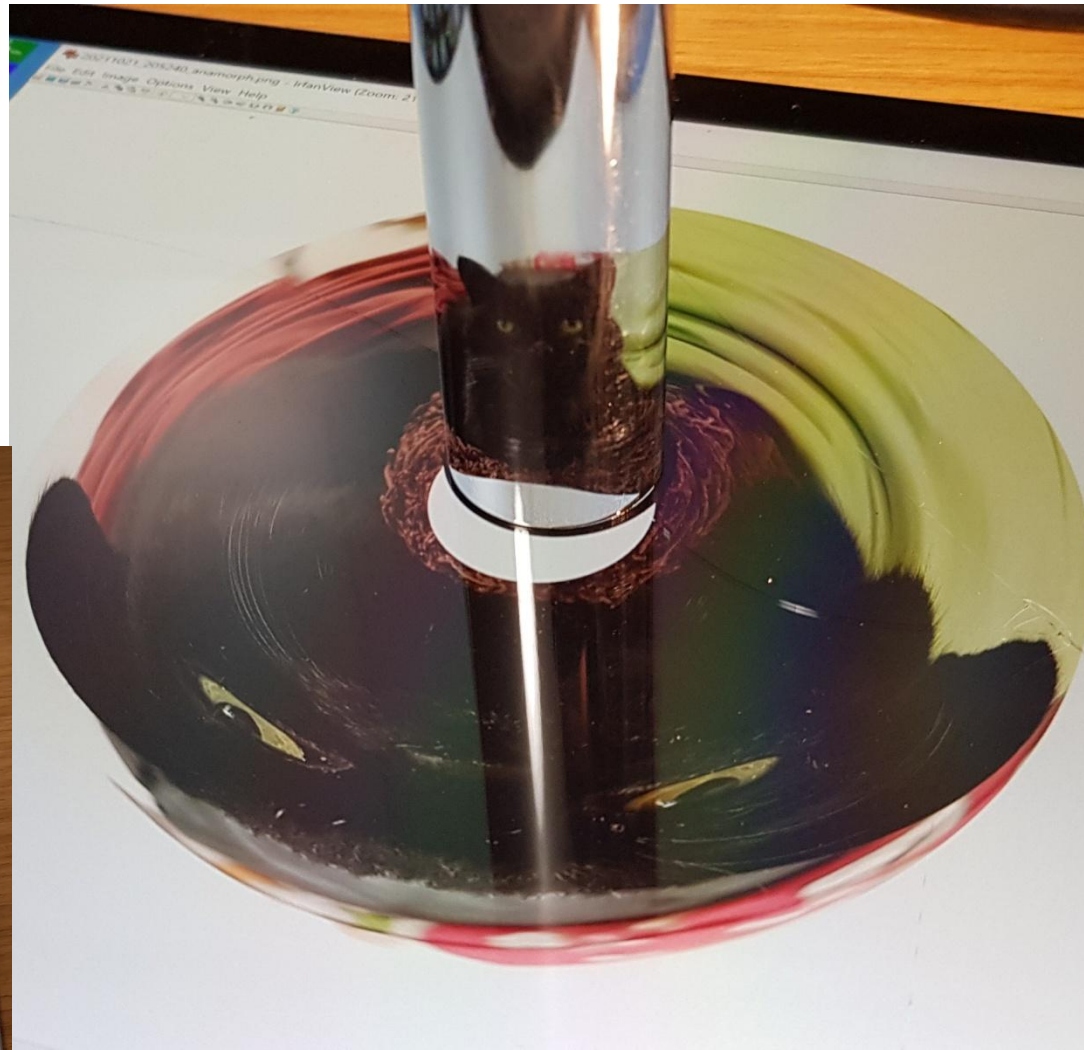
A 2.5cm diameter curtain rail section works as a mirror.

If you load the image onto a flat tablet screen and zoom until a polished cylinder fits into the black circle, the virtual image is of the correct proportions! You also don't have to print...

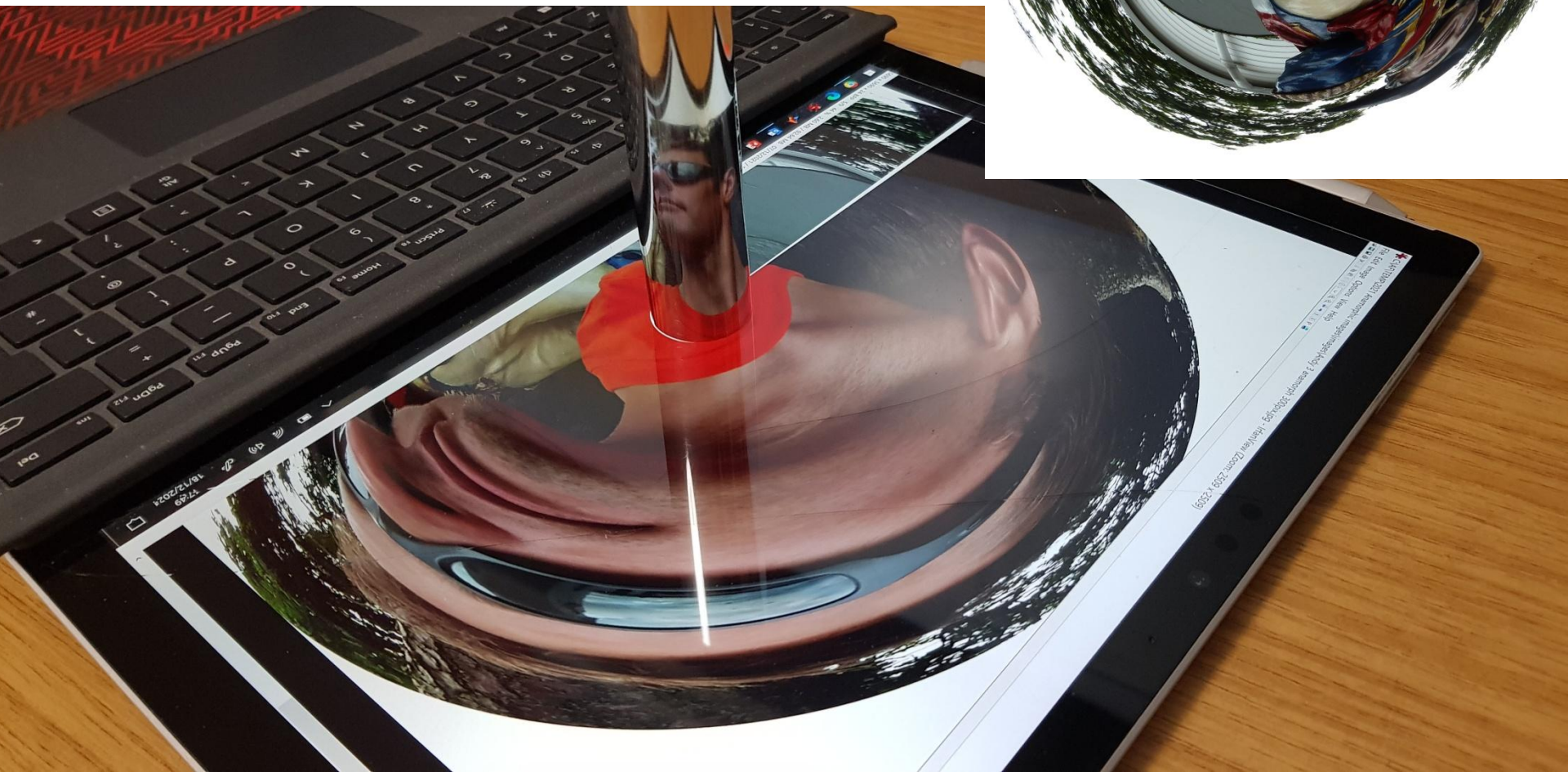


arc_deg = 160 Rf = 3

Sybil the cat was unperturbed by this anamorphic transformation.

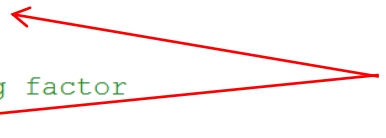


‘Fear and loathing’ in Portmeirion ...



MATLAB code
anamophic.m

```
%anamorphic
% Loads a bitmap image and fits this within a unit circle. An
% anamorphic projection is then determined such that reflection of the
% projection in a vertically oriented cylindrical mirror has the same
% proportions as the original 2D image, but appears to float in space.
%
% LAST UPDATED by Andy French Dec 2021.

function anamorphic

%Set opening angle of anamorphic arc /degrees
arc_deg = 300;

%Radial scaling factor
Rf = 3;

%Plot image in circle?
plot_image = 0;

%

%Import bitmap image of object and store RGB values in range 0...1 as a
%Iwidth x Iheight x 3 array
[filename,pathname] = uigetfile('*.*','Select bitmap to be anamorphed!');
if isempty(filename); return; end
I = imread([pathname,'\',filename]); I = double( flipdim(I,1) )/255;

%Get width and height in pixels
dim = size(I); Iwidth = dim(2); Iheight = dim(1);

%Scale image so it fits into a unit circle centred on the origin
sf = 0.5*sqrt( Iwidth^2 + Iheight^2 );

%Determine x,y coordinates of pixel locations
x = linspace(0,Iwidth/sf,Iwidth) - 0.5*Iwidth/sf;
y = linspace(0,Iheight/sf,Iheight) - 0.5*Iheight/sf ;
[x,y] = meshgrid(x,y);

```



Note these input parameters

```
38
39      %Create figure and plot a circle of radius R. If option set, plot original
40      %image within the circle. This is the untransformed original.
41 -    figure('name','anamorphic');
42 -    if plot_image==1; surf(x,y,-ones(Iheight,Iwidth),I); end
43 -    shading interp; view(2); hold on; axis equal; axis off;
44 -    theta = linspace( 0, 2*pi, 1000 );
45 -    plot( cos(theta), sin(theta), 'k-','linewidth',3 );
46
47      %Determine x,y coordinates of anamorphic grid, i.e. the pixel locations for
48      %the transformed image.
49 -    r = linspace( 1, Rf*Iheight/sf + 1, Iheight );
50 -    theta = linspace( -0.5*(arc_deg*pi/180), 0.5*(arc_deg*pi/180), Iwidth );
51 -    [theta,r] = meshgrid(theta,r);
52 -    x = r.*sin(theta); y = -0.5*Iheight/sf -r.*cos(theta);
53
54      %Plot anamorphic image
55 -    surf( x,y,-ones(Iheight,Iwidth),I); shading interp;
56
57      %Save 300DPI PNG image and close figure
58 -    if plot_image==1
59 -        filename = [pathname,'\',filename(1:end-4),'_anamorph_with_image.png'];
60 -    else
61 -        filename = [pathname,'\',filename(1:end-4),'_anamorph.png'];
62 -    end
63 -    print( gcf, filename,'-dpng','-r300' ); close(gcf);
64
65      %End of code
```

This is the anamorphic (x,y) original image to polar coordinate grid mapping.

Perhaps not
the most flattering
of examples

Festive greetings!