exercise 6

interpolation and image warping

solutions due

until **January 23, 2022** at **23:59** via **ecampus**

students handing in this solution set

last name	first name	student ID	enrolled with
Bach	Franziska	123456	B-IT / RWTH Aachen
Wolfe	Frank	654321	Uni Bonn

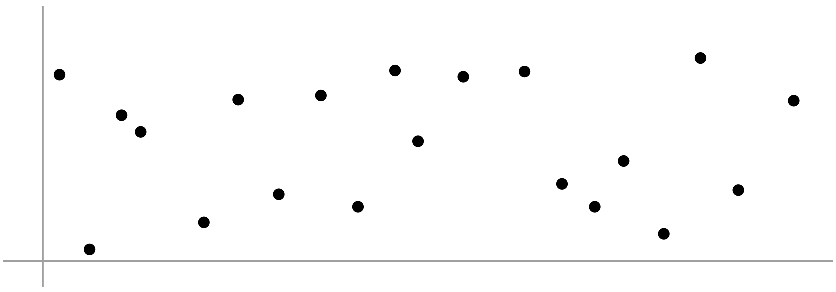
task 6.1

interpolating discrete uni-variate functions

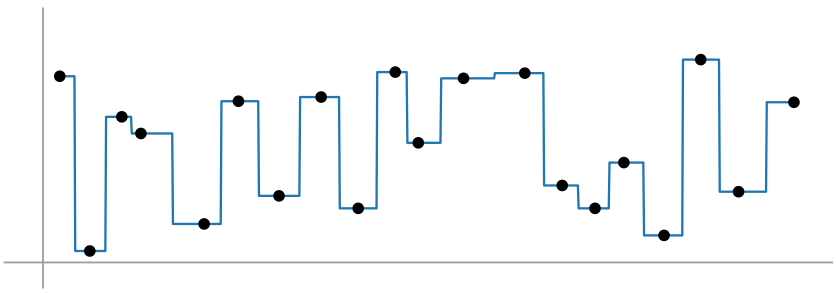
In this task, we briefly revisit the idea of interpolating discrete uni-variate functions. The following two arrays contain sample points x and function values $y = f[x]$

```
xs = [0.43, 1.20, 2.02, 2.51, 4.13, 5.01, 6.05, 7.13, 8.08, 9.03,
      9.62, 10.78, 12.35, 13.31, 14.15, 14.89, 15.92, 16.86, 17.83, 19.25]
ys = [1.79, 0.11, 1.40, 1.24, 0.37, 1.55, 0.64, 1.59, 0.52, 1.83,
      1.15, 1.77, 1.82, 0.74, 0.52, 0.96, 0.26, 1.95, 0.68, 1.54]
```

Note that the sample points are *not* equally spaced! If you plot this discrete function, your result should look something like this:



Now, use `xipl = np.linspace(np.min(xs),np.max(xs),1000)` to create an array of interpolation points and use what you have previously learned about the *scipy* function `interp1d` to interpolate the above function f as a function $\hat{f}$. For instance, if you were using nearest neighbor interpolation, a plot of f and $\hat{f}$ should look something like this:



Implement code that performs interpolation of discrete uni-variate functions and use *linear* interpolation and *cubic spline* interpolation to obtain two interpolated functions $\hat{f}$.

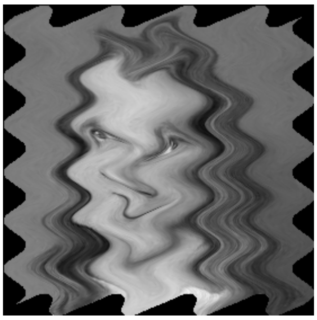
Create plots of your results and enter then here

put your figure here	put your figure here
linear interpolation	cubic spline interpolation

task 6.2

another kind of waves warp

Implement code that warps an input image into output images like these:


Hint: Recall what you have learned about wave functions. Which mathematical function featured prominently in this context? Which role could the parameters amplitude α, frequency ν, and phase ϕ play in the above pictures?

Run your code on the image `clock.jpg` and experiment with different parametrizations. Write your results as PNG files and enter three of them here

put your figure here

put your figure here

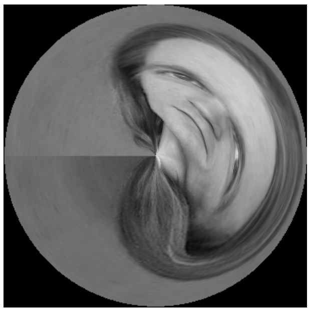
put your figure here

task 6.3

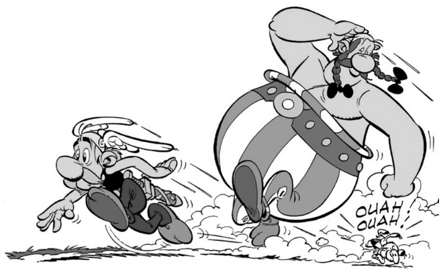
cylinder anamorphosis

Cylinder anamorphosis was a popular form of art in the 18th century where images were drawn in a warped fashion. They would appear un-warped when viewed in a cylindrical mirror.

a) Implement a program that can compute pictures such as these:



b) Next, extend your program such that it not confined to mapping images onto radial discs but can also map them onto a torus:



Apply your extended implementation to the image `flower.png` and experiment with different parameter choices. Write your results as PNG files and enter 4 of them here:

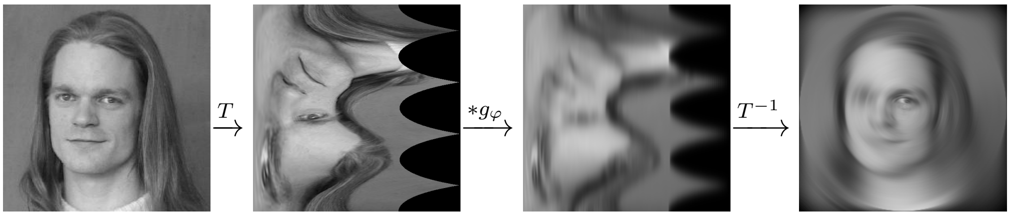
put your figure here	put your figure here	put your figure here	put your figure here

task 6.4

radial blur

In one of our early lectures, we discussed the idea of representing an image in $[r, \varphi]$ coordinates rather than in $[x, y]$ coordinates.

Later, we also saw, that interesting effects result from applying Gaussian smoothing to this representation. In particular, if we transform from $[x, y]$ to $[r, \varphi]$ coordinates, apply a 1D Gaussian filter along the φ dimension of the transformed image, and then transform the result back into $[x, y]$ coordinates, we obtain a circularly blurred image as seen below:



x, y image | r, φ image | φ-axis blurred | new x, y image

Implement a program that realizes this chain of processing steps. Apply it to the image `clock.jpg` and experiment with different choices for the variance parameter of the Gaussian filter kernel. Write four of your results as PNG files, and enter them here.

put your figure here | put your figure here | put your figure here | put your figure here

task 6.5

augmented reality

In the lecture, we discussed perspective mappings between quadrilaterals to create simple augmented reality effects like this:

Implement a program that maps the image `clock.jpg` onto the poster seen in the image `isle.jpg`. Assume that the four corners of the poster coincide with following $[u, v]$ array coordinates

$$\boldsymbol{u}_{ul} = \begin{bmatrix} 215 \\ 56 \end{bmatrix} \quad \boldsymbol{u}_{ur} = \begin{bmatrix} 365 \\ 10 \end{bmatrix} \quad \boldsymbol{u}_{ll} = \begin{bmatrix} 218 \\ 258 \end{bmatrix} \quad \boldsymbol{u}_{lr} = \begin{bmatrix} 364 \\ 296 \end{bmatrix}$$

where we assume that the origin $[0, 0]$ of the image `isle.jpg` is located in its upper left corner.

Write your result as a PNG file, and enter it here.

put your figure here