# Assignment 1 - Image Morphing

In this assignment you will be implementing the image morphing algorithm presented in the practical session. See attached paper and slides.

## Basic Implementation

Download the zip and run the executable. You should change the shader inside the shaders folder. You can add your images and your configuration instead src1,dst1 and cfg.

Configuration file in the following format:

line 1: c <sequence length> <a> <b> <p>

Following lines should contain pairs of corresponding lines in the source and destination images one after the other, defined by their starting and ending points:

s <x_src> <y_src> <x_src> <y_src>

d <x_dst> <y_dst> <x_dst> <y_dst>

<a>,<b> and <p> are the coefficients that control the influence of each line on the warping, as described in practical session #2 slide 15

**Implementation notes:**

1. You can assume that the source and destination images have the same shape (but different image pairs may have different shapes).
2. The warping algorithm might map some pixels in the destination image to coordinates outside the source image. In that case, put 0 (i.e. black) in the destination pixel.
3. Lines is giving to you by pixels (number between 1 to image width or height). You should convert them to texture coordinates (number between 0 to 1). Width and height of the image will be given in sizes[2] and sizes[3].
4. Lines will give to you in src_lines and dst_lines. You have to interpolate the lines using time (time*l[i] + (1-time)*l'[i]).
5. You may add functions. You can use uniform variables as global variables.
6. Shader function you may use: length, normalize, dot, pow. Be aware that * among vectors will give you a vector.
7. The distance dist[i] referenced in slide 15 is not a distance from a point to an infinite line, but rather from a point to a line segment (bounded by 2 end-points). You can figure out on your own how to calculate the correct distance, or you can look it up in the paper itself (page 3).
8. For creating your own morph sequences, you can use the line correspondence editor from http://www.cs.princeton.edu/courses/archive/spr11/cos426/morphlines.html. However, note that the y coordinates the editor produces are relative to the bottom of the image, while the order of the pixels when loading the bitmap file starts from the top.

## Submission

Submit a zip file with a name in the format <id1>_<id2>.zip, where <id1> and <id2> are your IDs. Include in the zip file only the shader you wrote (.glsl). Failure to compile and run your program will result in grade 0. You may appeal with a fix of the problem, but 20 points would be reduced from your assignment grade. Therefore, make sure that your program compiles and runs correctly, even on other computers.

The zip file should also include the readme document for the creative extras section, as well as any additional material needed.

Submission is in pairs (unless explicitly permitted by the course's staff).