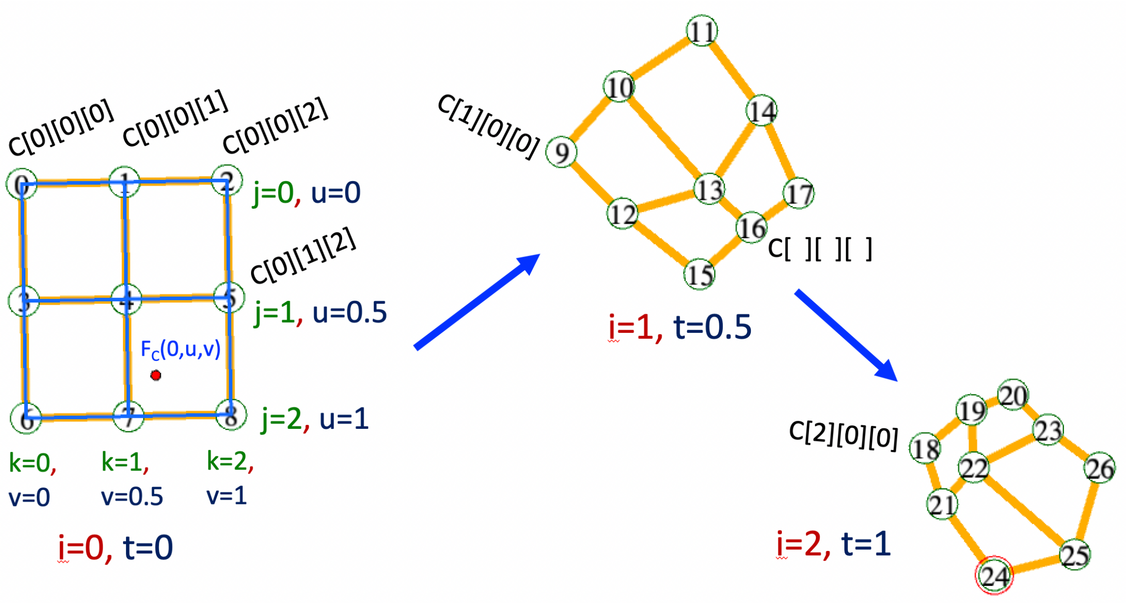
CS3451-2020-PROJECT 3: Animated Image Warp and Morph

1 Problem statement

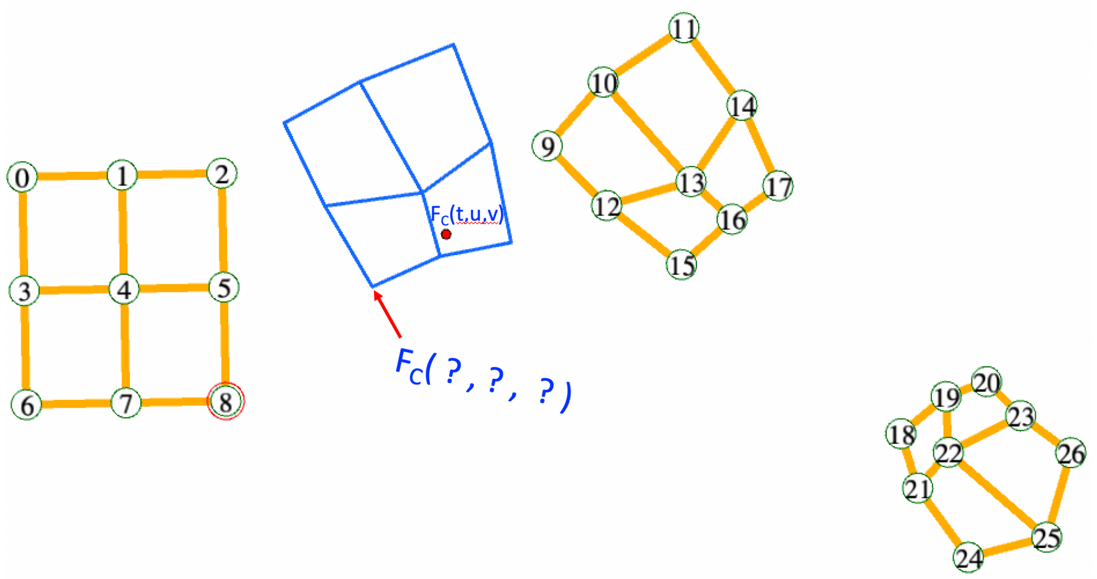
Define $F_C(t, u, v)$, a <u>continuous</u> and <u>smooth</u> <u>point-valued</u> function of three variables that interpolates 27 constraints:

$F_C(i/2, j/2, k/2) = C[i][j][k]$ with $i,j,k \in \{0,1,2\}$,

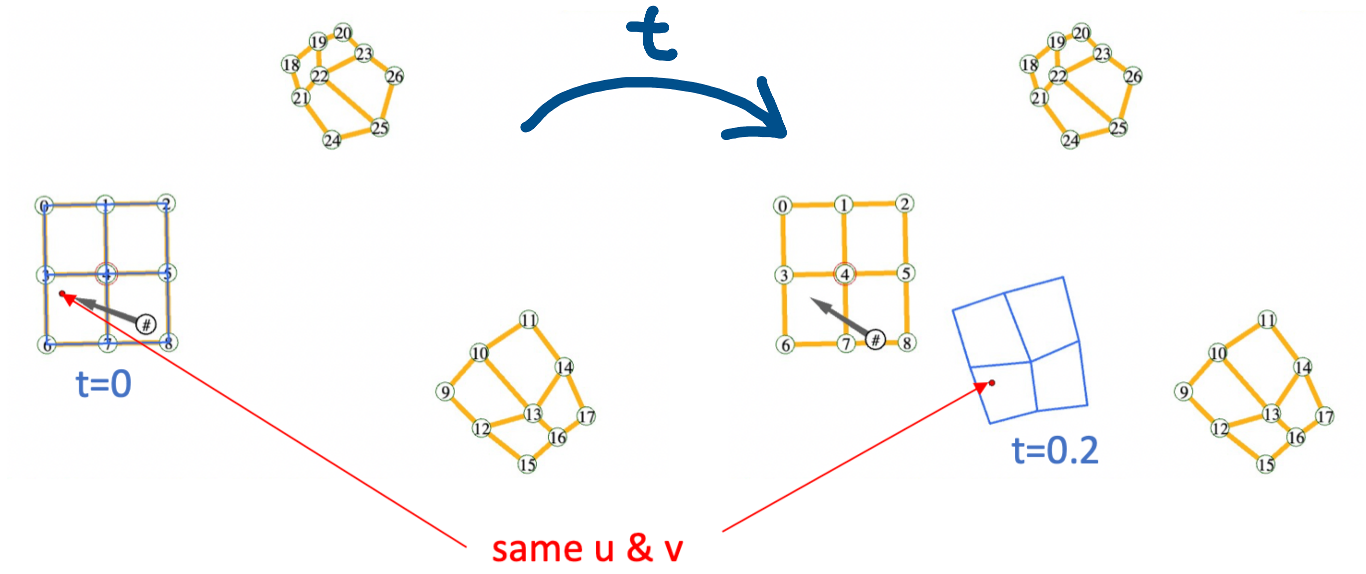
where PNT C[3][3][3] is the array of control points for F_C.



For any given value of $t \in [0,1]$, $F_C(t,u,v)$ defines a point (red).

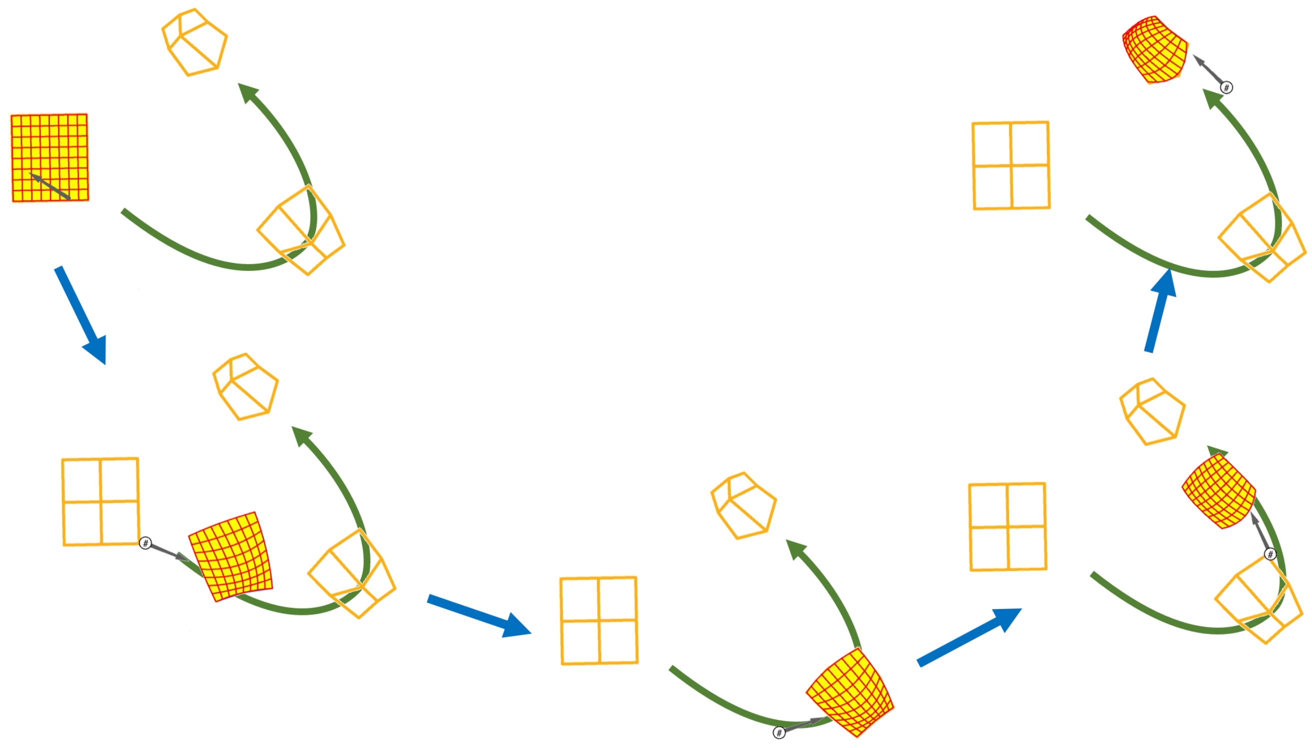


Define (u,v) with the mouse (relative to the whole canvas) and track it over-time to ensure that your implementation of Fc is correct.

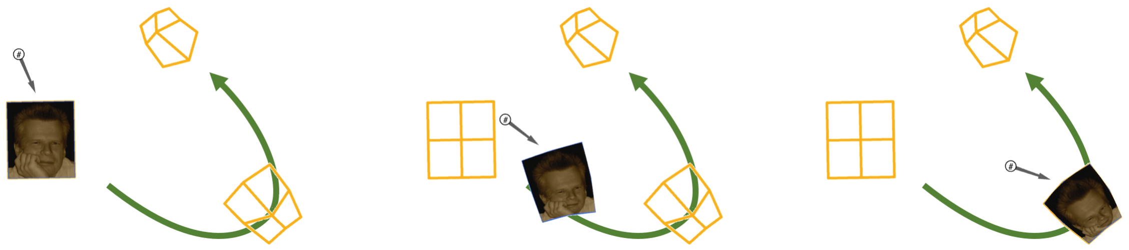


For any given value of time t ∈ [0,1], Fc(t,u,v) defines the array PNT G[n+1][n+1] of points G[j][k] = Fc (t , j/n , k/n),
which are the images by Fc of the regular grid R of points (j/n,k/n) in the (u,v) parameter space.

Compute G[][] and use it to display a lattice (yellow with red borders) of n×n quadrilateral tiles (each being the image by Fc of a square of R.

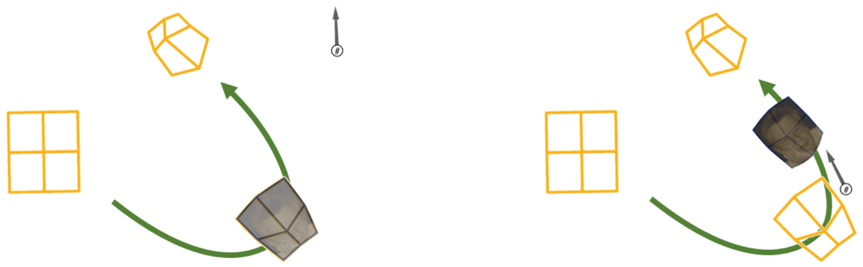


Use these tiles to texture map a picture P_0. During animation (as t changes from 0 to 1), this image warp evolves smoothly and interpolates exactly the 3 keyframes at the corresponding times.



Show how, by tweaking the 3×3 grids, C[0], C[1], C[2], the artist can control the 3 keyframes (distorted versions of P_0) associated with times 0, ½, and 1.

Add the option to make the texture appear progressively through the animation and use this to cross-fade when morphing between two images.



2 Phase A (individual): Tri-Quadratic Warp Animation

Use the Processing sketch provided. Implement $F_C(t, u, v)$ using **parabolic interpolants** with knots 0, ½, and 1.

Demonstrate it: by animating the edges of the quads of G and by animating a warp of your face.

Provide the option for fade-in (using TINT).

Submit (the usual) slides stating the problem, outlining the solution, providing the math formulae (using points & vectors, not coordinates), list of sources of inspiration, assumptions and limitations, an image of your code, images, a GIF, and a zip of your implementation.

3 Phase B (team): Log-Spiral Warp Animation

Implement a version of $F_C(t, u, v)$ using **log-spiral interpolants** and a key ('L') allowing the user to toggle between parabolic and log-spiral. Write on canvas which one is shown. Provide the usual slides for this solution and include a slide or two comparing the two formulations. (Discuss which one appears smoother? Which one is easier—more intuitive—to use? Why?)

4 Phase C (team): Image Morph

- Implement, explain (using slides) and show (GIFs) one of these two applications:
  1) Morph between front-facing pictures of faces of two **different** people. Use the keyframes to align their features and crossfade (TINT) between one warp-animation and the inverse of the other to achieve a morph.
  2) Morph between two pictures (or interpolate through 3 pictures) of the **same** face taken from **different angles**.
- Invent and implement a GUI that helps with the feature alignment in one of the applications above. For example, let the user click on the eyes and mouth corners in the images. Explain what is hard about it, evaluate your solution, discuss its limitations, suggest further research.

5 Schedule and delivery

One week per phase. Phase 1 due Sept, 22. Submit phase 2 and 3 together (Oct 6). Both due on Tuesdays **before** class.

Up to 3 days extension, with 5% (of full grade) penalty per day.

In your slides and code, clearly indicate (cover page of slides and on the canvas of your sketch) who are the members of your team.

Make sure that you submit the math details for each solution and some explanations showing that you understand what it does and why.