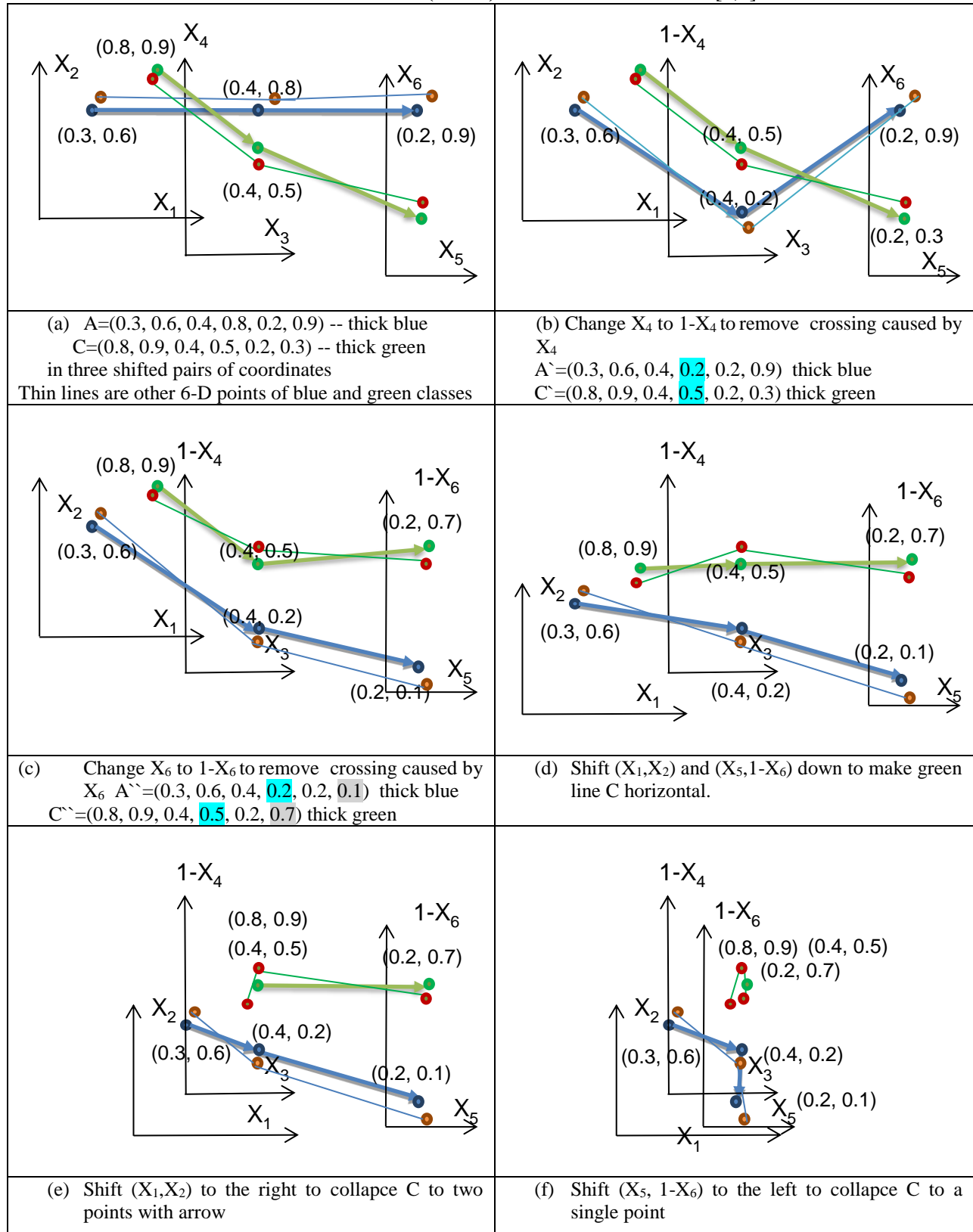
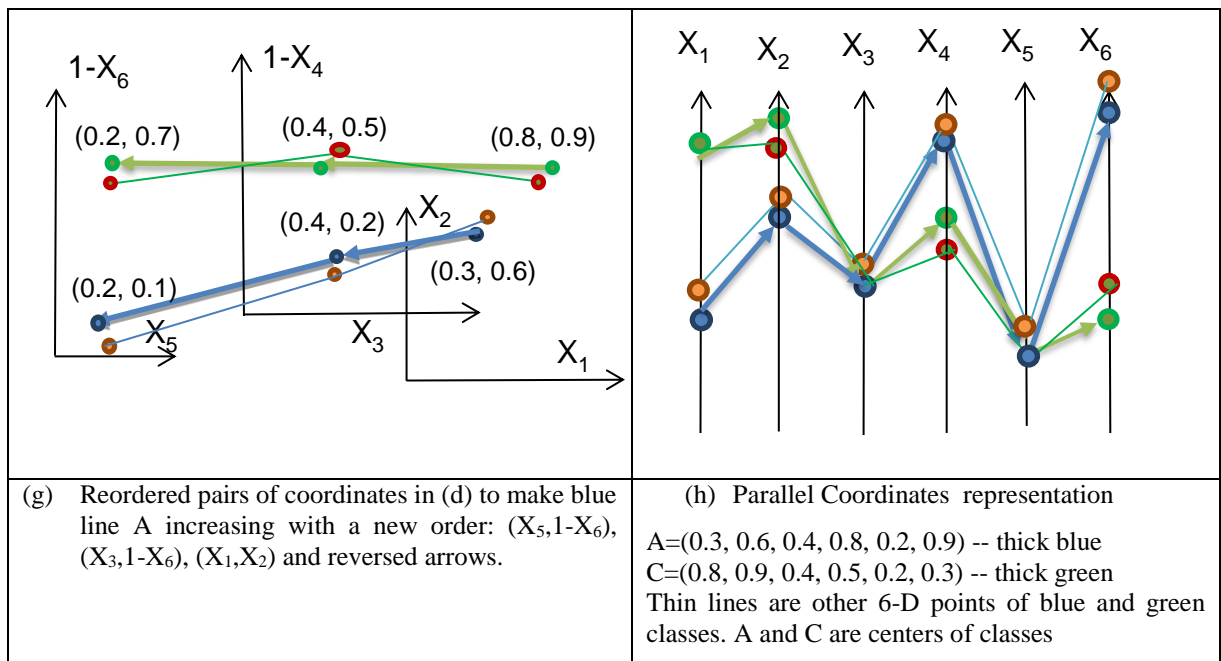


General Line Coordinates

Parameterized Shifted Paired Coordinates (PSPC) All values are scaled to $[0,1]$





HW2

Part 1

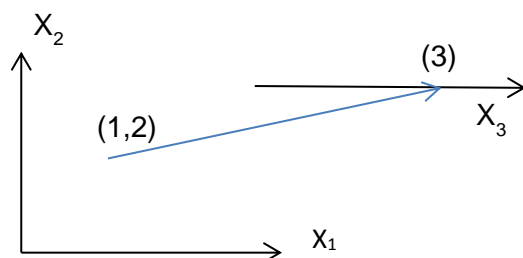
- Write code to produce figures (a)-(g) and draw figure (h) using code from HW1. All data must be entered from the file. Lines A and C must be made wide lines by clicking on the right button on them and return to the thin lines by click on the left button on them. Show arrows and coordinate labels. You can omit circles. Drawing numbers such as (0.2,0.1) is not required. See tips on text in OpenGL below. The transition from figure (a) to (b) must be done by clicking on b on the keyboard. To move from (b) to (c) click on c and from (c) to (d) click on d. From (d) must be options to go to (e),(f) or (h). From (g) must be options to go to (e) or (f).

- Generalize code (1) to $n=10$ and 100 cases of 3 classes colored blue, green and red. Read 100 cases from the file.

Part 2

- Conduct computational experiments with three datasets with 100 cases each with 2-3 classes colored blue, green and red. Read those 100 cases from the file. At least one dataset must have 3 classes.

Tip: for odd n , e.g. $n=3$ you will have a pair of coordinates (X_1, X_2) and X_3 as a separate coordinate, as shown in figure below for the point (1,2,3)



TEXT IN OPENGL

<https://www.opengl.org/archives/resources/features/fontsurvey/>

GLUT

There are two options for drawing fonts in GLUT:

- Bitmap - `glutBitmapCharacter()`
- Stroke - `glutStrokeCharacter()`

Advantages

- Very easy to use.

Disadvantages

- Limited in choice of fonts. Hard to add new fonts.
- What if you're not using GLUT? Then you have to link with GLUT.

Example Program

- Requires [GLUT](#).
- Download (Unix) - [glutfonts.tar.gz](#).
- Download (Windows) - [glutfonts.zip](#).

Example Output

