

CS 3220 Spring 2010 Homework 4

This homework includes a written and a MATLAB component, and it concerns matrix transformations of the plane.

The framework for this assignment is a MATLAB function that displays a picture twice, once on each side of the window, and allows the user to select three points on each side by moving handles around with the mouse (Figure 1). The picture on the right is the same as on the left but transformed by an affine transformation, and it is your job to compute the transformation so that the picture will follow the handles.

In particular, you should implement the function `computeLinearTransformation` to return a matrix that transforms the handles' left-hand locations (in the argument `origPts`) to the handles' right-hand locations (in the argument `newPts`). The framework calls this function whenever the handles are moved to update the matrix it uses to transform the picture.

Your function should find the matrix \mathbf{M} such that

$$\mathbf{M}\mathbf{q}_i = \mathbf{p}_i \quad \text{for } i = 1 \dots 3.$$

where the \mathbf{p} s are the right-hand handle positions and the \mathbf{q} s are the left-hand positions. Even though the problem is about the two-dimensional plane, we can't just use 2-vectors and a 2×2 matrix, because a linear transformation cannot move the origin: wherever $(0, 0)$ is in the picture, that point would end up being fixed. Instead we use 3-vectors for \mathbf{p} and \mathbf{q} ; we add a 1 to the end of each:

$$\mathbf{p} = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}$$

This way, when we transform the points by a 3×3 matrix, we can add a constant to the x or y coordinate by placing nonzero values in the third column of \mathbf{M} .

Problem 1: Explain how you can compute \mathbf{M} by solving a multi-RHS linear system.

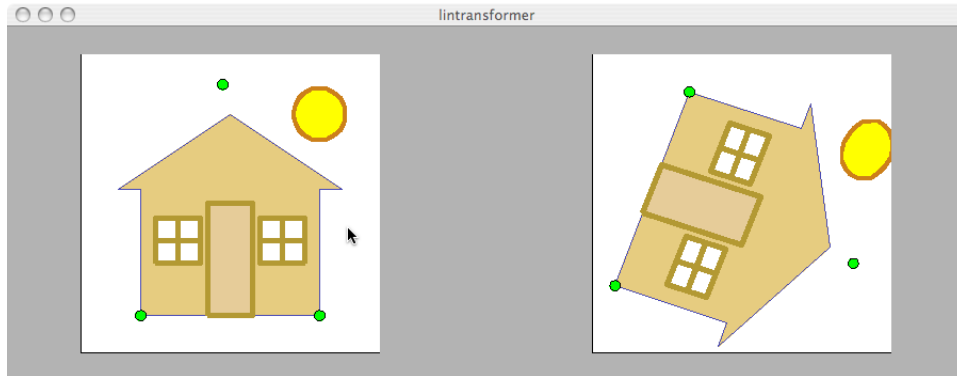


Figure 1: A screenshot of the program in action

Problem 2: Implement LU factorization without pivoting and use it to solve for \mathbf{M} in `computeLinearTransformation`. Be sure to compute the L and U factors just once, and then use them to solve for all three right-hand sides. When your code works, you should find that the house moves with the handles. Comment your code so that we can understand what it does. Also hand in the console result by running the test script `testLU`.

Problem 3: For what configurations of the control points is your function unable to work? Explain the two kinds of input that can cause your code can fail. If you instead calculate the linear transformation with MATLAB's "`\`" operator, which configurations still fail?