Pset3

4.1)

a. You need 2 points. This is because each point contains two numbers, and the total degrees of freedom in 2D is 3. Therefore two points, containing a total of 4 numbers, is enough to constrain all 3 DOF because 4 > 3.

b. You need three points for similar reasons. Each point in 3D carries 3 numbers, and their are 7 DOF in 3D space. Therefore you need at least three points, or nine numbers, to constrain all 7 DOF.

4.4)

a. The decision variables would be the translational component of a 2D transform since we know the rotation to be already correct. This means we are searching for an x and y that when added to the scene points make the two shapes overlap.

b. The decision variables do not show up in the constraints since the translation part of a transformation could be infinite in either x or y. This is unlike the rotation matrix which could also be a part of the transformation we are solving for which must be constrained to be a valid rotation matrix.

c. The objective function is the sum of the square of the L2 norms of the relative position between corresponding points.

For P_O = [0 0], sum = 327

For P_O = [3 10], sum = 0

For P_O = [6 12], sum = 39

d. It would be quadratic with respect to the x and y of P_O. This means the problem is convex and can be solved quickly and a global optimum will be reached.

```
import numpy as np


def obj(p_s, p_m):
    s = 0
    for i in range(len(p_s)):
        s += np.linalg.norm(p_s[i] - p_m[i]) ** 2
    return s


p_s = np.array([[1,5],
                [3,10],
                [5,10]])


p_om = np.array([[-2,-5],
                [0,0],
                [2,0]])


p_o = np.array([[0,0],
                [3,10],
                [6,12]])


for p in p_o:
    print(p)
    print(obj(p_s, p_om + p))
```

4.5) Done on paper
4.8) Paper

Survey: Meshcat Mustard