Workspace 'pset4 geometry I' in '6.4212 Manipulation'
Page 1 (row 1, column 1)

Worked w/ Arthur Pommersheim, Marine Maisonneuve, + April Hu

4.1 a)
To uniquely specify the pose of an object in 2D with known correspondences, we need only two points to fully specify the object. The object position and rotation in 2D is defined by 3 DOF: xy position and theta rotation; we need solutions to three variables to fully define the solution. Consider the simplest possible 2D object with non-zero volume: a triangle. If we know the correspondences, we intrinsically know that we have not misrepresented the rotation of the triangle because the correct points are mapped together (i.e. not reached a local minimum), but we still have to fully define the location of the object in all three degrees of freedom.

4.1 b)
In 3D, the problem becomes slightly more complicated and requires a fourth point. Consider once again the simplest 3D object, a tetrahedron or pyramid. We have three degrees of translational freedom and three rotation axes. All four points of the object must be defined to fully define the pose of the object. Even for more complex objects, however, the overall pose can still be defined by those four points.

4.4 a) ${}^Wp^{s_0} = (1,5)$ ${}^Op^{m_0} = (-2,-5)$
${}^Wp^{s_1} = (3,10)$ ${}^Op^{m_1} = (0,0)$ $R^O = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
${}^Wp^{s_2} = (5,10)$ ${}^Op^{m_2} = (2,0)$

Decision vars are ${}^Wp^{O_x}$ & ${}^Wp^{O_y}$. These are the x & y coordinates of the transform ${}^WX^O$.

4.4 b) $\underline{p^O = (0,0)} \rightarrow$ no transform, so sum-squares is

$(1--2)^2 + (5--5)^2 + (3-0)^2 + (10-0)^2 + (5-2)^2 + (10-0)^2$
$= 9 + 100 + 9 + 100 + 9 + 100$
$\boxed{= 327}$

$\underline{p^O = (3,10)} \rightarrow$ apply transform to ${}^Op^{m_0}, {}^Op^{m_1}, {}^Op^{m_2}$

so now: ${}^Wp^{m_0} = (1,5)$, ${}^Wp^{m_1} = (3,10)$, ${}^Wp^{m_2} = (5,10)$ — these match the ${}^WX^S$ transform points, so the objective function is 0.

$(1-1)^2 + (5-5)^2 + \ldots$ etc $= \boxed{0}$

$\underline{p^O = (6,12)} \rightarrow$ so $p^{m_0} = (4,7)$; $p^{m_1} = (6,12)$; $p^{m_2} = (8,12)$

$= (1-4)^2 + (5-7)^2 + (3-6)^2 + (10-12)^2 + (5-8)^2 + (10-12)^2$
$= 9 + 4 + 9 + 4 + 9 + 4$
$= \boxed{39}$

4.4 c)
Each of these terms forms a parabola individually, so the combined shape would be a paraboloid, or a parabola rotated about the z axis (if the value of the objective function is portrayed in z). This result is the same as that of only varying the rotation as seen in Example 4.2. This result is significant because it enables a convex optimization yielding no local minima and one global solution which can be computed quickly and reliably.

4.4 d) To solve this in the general case, we want to run the optimization's objective function until the objective value is no longer changing. Thus, we can differentiate it w/rt each variable (Lagrangian).

obj. fn.: $\min_{p,R} \sum_{i=1}^{N_s} \| p + R^O p^{m_{ci}} - p^{s_i} \|^2$ s.t. $RR^T = I$

Lagrangian: $\sum_{i=1}^{N_s} 2(p + R^O p^{m_{ci}} - p^{s_i}) = 0$

$p^* = \frac{1}{N_s} \sum_{i=1}^{N_s} p^{s_i} - R^* \left(\frac{1}{N_s} \sum_{i=1}^{N_s} {}^Op^{m_{ci}} \right)$

since Rotation is fixed, $R^* = R^O$.

so $\sum_{i=1}^{N_s} 2(p + R^O p^{m_{ci}} - p^{s_i}) = 0$

$N_s \cdot p + \sum_{i=1}^{N_s} (R^O p^{m_{ci}} - p^{s_i}) = 0$

solve for p

$\boxed{\text{so } p = \frac{1}{N_s} \sum_{i=1}^{N_s} (p^{s_i} - R^O p^{m_{ci}})}$

${}^Wp^O = \sum_{i=1}^{N} \| p^{s_i} - (p^{m_i} + {}^Wp^{O_x}) \|^2 \rightarrow$ sum-square differences of each coordinate

so $p_x = \frac{1}{N} \sum_{i=1}^{N} (x_{si} - x_{mi}) \rightarrow$ which is the avg posn of all point cloud dif in X.

similarly, $p_y = \frac{1}{N} \sum_{i=1}^{N} (y_{si} - y_{mi})$ so $p^* = \frac{1}{N_s} \sum_{i=1}^{N_s} ({}^Wp^{s_i} - {}^Op^{m_i})$

This method finds an optimal solution even if there's noise.

4.5 a) $\min_{p_x, p_y, a, b} \sum_i \left\| \begin{pmatrix} p_x \\ p_y \end{pmatrix} + \begin{pmatrix} a & -b \\ b & a \end{pmatrix} {}^Op^{m_{ci}} - {}^Wp^{s_i} \right\|^2$ s.t. $a^2 + b^2 = 1$

plug in for $p_x = 0$, $p_y = 0$, $a = 1$, $b = 0$

$\sum \left\| \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} {}^Op^{m_{ci}} - {}^Wp^{s_i} \right\|^2$

ferences

so now the summation becomes $\sum \left\| \begin{pmatrix} 0 \\ 0 \end{pmatrix} + {}^{O}p^{m_{ci}} - {}^{W}p^{s_i} \right\|^2$

if our initial guess is right, then ${}^{O}p^{m_{ci}}$ aligns w/ ${}^{W}p^{s_i}$ $\forall i$. All the points subtract to 0.

So $\text{cost} = \sum_i \left\| \begin{pmatrix} 0 \\ 0 \end{pmatrix} \right\|^2 \rightarrow$ ICP cost is 0. All the points are correctly placed and the correspondences align.

The set of initial poses where ICP converges to the correct solution is given by cases where the values of *a* in the rotation matrix above are positive. Given a fixed rotation where the rotation of the scene pose matches that of the model pose, we can assume that ICP will always produce the correct correspondences. Therefore, translation does not effect correespondence matching. If abs(rotation) about theta is greater than pi/2, the shape is rotated such that the correspondences will invert. This affects the cos(theta) terms of the 2D rotation matrix; in cases where abs(theta) > pi/2, the value of cos(theta) will be negative, driving incorrect correspondences. So *a* must be positive.

b) $\sum_i \left\| \begin{pmatrix} p_x \\ p_y \end{pmatrix} + \begin{pmatrix} a & -b \\ b & a \end{pmatrix} {}^{O}p^{m_{ci}} - {}^{W}p^{s_i} \right\|^2$

$p_x, p_y = 0,\ a = -1,\ b = 0$

plug-in

$\sum_i \left\| \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} {}^{O}p^{m_{ci}} - {}^{W}p^{s_i} \right\|^2$

this inverts both terms.

$\sum_i \left\| \begin{pmatrix} 0 \\ 0 \end{pmatrix} - {}^{O}p^{m_{ci}} - {}^{W}p^{s_i} \right\|^2 \rightarrow$ as we solve the summation and minimize the objective function, we find that ${}^{O}p^{m_{ci}} = -{}^{W}p^{s_i}$! So the correspondences are flipped.

As the corollary to 4.5a), this incorrect solution develops in cases where the rotation of the scene w/r/t the model is rotated greater than pi/2. Put another way if scene rotation is defined with angle theta, abs(theta) > pi/2 leads to incorrect pose solutions during the solving process. The algorithm will take the direction of shortest rotation, producing the incorrect estimation.

4.5 c) $W = \sum_{i,j} c_{ij} (p^{s_i} - p^{\bar{s}})({}^{O}p^{m_j} - {}^{O}p^{\bar{m}})^T$.

say ${}^{W}p^{s_1}$ and ${}^{W}p^{s_2}$ both correspond to ${}^{O}p^{m_1}$. We can compute ${}^{O}p^{\bar{m}}$ as $\frac{1}{2}\left(\underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{{}^{O}p^{m_1}} + \underbrace{\begin{pmatrix} -1 \\ 0 \end{pmatrix}}_{{}^{O}p^{m_2}}\right) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

likewise, we find the centroid of scene points. We compute ${}^{W}p^{\bar{s}}$ as $\frac{1}{2}\left(\underbrace{\begin{pmatrix} 1 \\ 0 \end{pmatrix}}_{{}^{W}p^{s_1}} + \underbrace{\begin{pmatrix} -1 \\ 0 \end{pmatrix}}_{{}^{W}p^{s_2}}\right) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$.

$W = \sum_{i=1}^{N_s} (p^{s_i} - p^{\bar{s}})({}^{O}p^{m_{ci}} - {}^{O}p^{\bar{m}})^T$

So $W = \underbrace{\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)^T}_{i=1} + \underbrace{\left(\begin{pmatrix} -1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)\left(\begin{pmatrix} -1 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix}\right)}_{i=2}$

$W = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \rightarrow$ perform SVD to get R^*

$R^* = UDV^T \rightarrow$ from the hint, SVD leads to I for U and V.

$R^* = I \cdot I \cdot I^T = I$ $\quad p^* = p^{\bar{s}} - \cancel{R^*}\,{}^{O}p^{\bar{m}}$

$p^* = p^{\bar{s}} - {}^{O}p^{\bar{m}} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$

begin ICP: ${}^{O}p^{m_1}$: $p^* + R^*\,{}^{O}p^{m_1} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + I\begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$

${}^{O}p^{m_2} \Rightarrow \begin{pmatrix} 0 \\ 0 \end{pmatrix} + I\begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$

sum-squared dist $= \boxed{2} \rightarrow$ nonzero cost.

In the next iteration, even though we recompute nearest neighbors, since the points are both trying to approach the same model point, the cost drops no further and the ICP algo gets stuck w/ cost > 0.

4.8 a) 1. six points: like nearest_neighbors, we can compute a closest model point $b_i \in \{i = 1 \dots 4\}$ for all scene points

$(b_1, r_1);\ (b_2, r_2);\ (b_2, r_3);\ (b_3, r_4);\ (b_3, r_5);\ (b_4, r_6)$ $\quad r_j \in \{j = 1 \dots 6\}$

the ICP error $= \sum \| (b_i - r_j) \|^2$ $\quad i = \{1 \dots 4\}$ $j = \{1 \dots 6\}$

sum distance-squared $= 2 + 16 + 2 + 2 + 17 + 2 = \boxed{41}$

2. Evaluating 6 scene points, I would expect 2 points to be outliers.

we can expect the two points farthest from their nearest-neighbor correspondant are outliers.

These are the pairs w/ sum-squared distances of 16 and 17 calc'd above.

Therefore, we can expect $\boxed{r_2 \text{ and } r_5 \text{ are outliers.}}$

3. New ICP = sum-squared distances ignoring distances attributed to r_2 & r_5.

Error: $ICP_{NEW} = ICP_{OLD} - \text{outliers} = 41 - (16+17)$
$= \boxed{8.}$

4.8 b) 1. b_4 is closest to r_5 w/ distance = 4.

2. r_1 is closest to b_1 w/ distance $= \sqrt{2}$.

3. Scheme 2 is more robust here; since outliers occur in the scene rather than in the model, there's some chance in scheme 2 that we are given an outlier. However, picking a model point means that we'll start out mapping w/ a true model point, leading to better performance in this particular case.

Survey Code: "point cloud"