Programming Assignment 4

Zhiyuan Ding zding20@jh.edu, Deming(Remus) Li dli90@jh.edu

November 26, 2021

1 Math approaches

1.1 Find closest point for a triangle and a point

Here we illustrate the computation process to find the nearest point for point a and triangle with vertices $[p, q, r]$. There are four different regions as in Figure 1. We first solve the least square for the relationship:

$$a - p \approx \lambda(q - p) + \mu(r - p)$$

and depend on λ and μ we get these four following conditions:

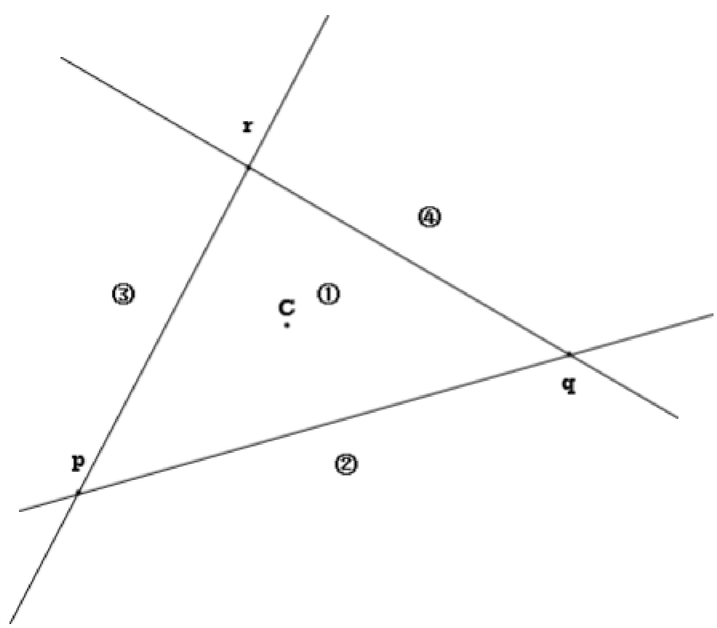

Figure 1: Four different regions in find closest point algorithm

- $\lambda + \mu \leq 1, \lambda > 0, \mu > 0$. Region 1 the projection of c lies in the triangle pqr. The nearest point is just point c as

  $$c = p + \lambda(q - p) + \mu(r - p)$$

- $\mu \leq 0$. The projection of a is c which lies below l_{pq} in Region 2. λ is computed as:

  $$\lambda = \frac{(c - p)(q - p)}{(q - p)(q - p)}$$

  Then the closest point $c^* = p + \lambda_{seg}(q - p)$, where $\lambda_{seg} = max(0, min(\lambda, 1))$.

- $\lambda \leq 0$. The projection of a is c which lies in the left side of l_{rp} in Region 3. λ is computed as:

$$\lambda = \frac{(c-r)(p-r)}{(p-r)(p-r)}$$

  Then the closest point $c^* = r + \lambda_{seg}(p-r)$, where $\lambda_{seg} = max(0, min(\lambda, 1))$.

- $\lambda + \mu \geq 1$. The projection of a is c which lies in the right side of l_{rq} in Region 3. λ is computed as:

$$\lambda = \frac{(c-q)(r-q)}{(r-q)(r-q)}$$

  Then the closest point $c^* = q + \lambda_{seg}(r-q)$, where $\lambda_{seg} = max(0, min(\lambda, 1))$.

1.2 Simple search with bounding box

Simple search with a bounding box generally filters out some points based on the distance between the cuboid containing the triangle and the object point.

Specifically, for a given bound, we need to check xyz-coordinates separately for:

$$L_x - bound \leq c_{k,x} \leq U_x + bound L_y - bound \leq c_{k,y} \leq U_y + bound L_z - bound \leq c_{k,z} \leq U_z + bound$$

, then with triangle holds the above requirement, we compute the nearest point in the triangle and use the computed point to update bound. This method is generally a linear searching method with time complexity $O(N)$

1.3 ICP method

In this programming assignment, main issue is to construct iterated closest point algorithm so that we can find a frame transformation for point registration and point set on the surface corresponding to the closest point.Specifically, the process can be divided into initialization, point registration, frame update and termination judgement parts. Given the original point set $\{q_i|i = 1, ..., k\}$, maximal iteration N,an threshold for outliers η_0 ICP can be described as the following process:

- In initialization part, the main task is to give an initial guess of frame transformation F_0 and build a data structure for closest point searching. After that a closest point set$\{c_i|i = 1, ..., k\}$ on the surface for current transformed point set and corresponding distance set $\{d_i|d_i = ||F_0 q_i - c_i||\}$.

- In point registration part, i.e. matching part, for the transformed point $F_n q_i$, closest point c_i on the surface is found with FindClosestPoint function we construct in PA3. Then the distance $\{d_i|d_i = ||F_n q_i - c_i||\}$ is computed and a judgment between η_n and d_i is made. If d_i is larger than η_n, the point pair $(F_n q_i, c_i)$ is seen as outliers and excludes from the following frame update part.

- In the frame update part, we use preserved point pairs $(F_n q_i, c_i)$ to proceed rigid frame transformation to obtain the updated frame F_{n+1}. This part can be done based on our PA1 functions RigidTransformation. And the residual error $e_i$is computed for threshold update. In this program, we use $\eta_{n+1} = 3\epsilon_n = \frac{\sum |e_i|}{k}$ as update criterion.

- In termination judgment part, we apply an early stopping strategy to check if we can stop the iteration earlier than maximal iteration reaches. Specifically, we check the ratio $\frac{\epsilon_n}{\epsilon_{n-1}}$ whether larger than a given parameter γ, which means that current iteration doesn't decrease the error a lot and has subtle effect on frame. If $\frac{\epsilon_n}{\epsilon_{n-1}}$ larger than γ for several iterations, then we stop the iteration before reaching maximal iteration.

2 Algorithms steps

In this part we show the algorithm step for ICP approach, as in 1. The main part is an iteration which finds the closest point on the surface and a rigid frame transformation. Three judgments are done to exclude the outliers, check the error changes and decide whether to stop the iteration before maximal iteration reaches.

Algorithm 1 Designed ICP algorithm

Input: maximal iteration N, Initial frame guess F_0,
Input: Surface S, point set $\{q_i|i = 1, ..., k\}$
Input: Initial threshold η_0, termination iteration N_T,$N_t = 0$ and γ
for $n = 1$ to N **do**
  A=[ ];B=[ ]
  Find closest point c_i on the surface S for $F_{n-1}q_i$ based on FindClosestPoint function.
  Compute $\{d_i|d_i = ||F_n q_i - c_i||\}$
  if $d_i < \eta_{n-1}$ **then**
    put q_i into A, put c_i into B
  end if
  a point rigid registration between A,B is done based on point set A, B to get F_{n+1}
  compute the registration residual error $e_i = |c_i - F_{n+1}q_i|$ and mean value $\epsilon_n = mean(e_i)$
  if $\gamma \leq \frac{\epsilon_n}{\epsilon_{n-1}}$ **then**
    $N_t = +1$
  end if
  if $N_T \leq N_t$ **then**
    end the iteration
  end if
end for
Output: Frame transformation F, closest point c_i

3 Overview of the program

Most of the part in PA4 is the same as PA3 and the only difference lies in utilizing an ICP methods to find F_{reg}, as in Figure 2.

First, the surface file is loaded and we can obtain the indices for triangle vertices and the corresponding coordinates.

Second, a rigid transformation is done to get the coordinate of A tip corresponding to each frame, which is solved by given files.

Finally, ICP method is deployed to get an estimation F_{reg} and corresponding closest point set. Two important subfunction we used in ICP construction is FindClosestPoint function which is built in PA3 and the PointRigidRegistration which is constructed in PA1.

4 Verification

In this part, we did two tasks to verify our program: In the first part, we plot a figure of transformation error to ensure that our ICP function can actually make the residual error decrease based on the update frame. In the second part, we measure the mean error for both transformed original debug data set and the found closest point compared with given answers.

4.1 Plot of error

We first plot the curve of transformation mean error to verify that update actually makes the error decreasing. We use the mean l2 error of rigid registration to represent it and get Figure 3.

We can see that the mean error decreasing with the iteration increases and get flat gradually. In fact, in this experiment we set the maximal iteration is 50 and at 24 iteration the error increase has been small enough for several iterations. Thus the early stopping condition is triggered.

In this experiment we ensure our programs actually generate a frame make the registration reasonable.

4.2 Mean error compared with known result

In this part, we verify our result by comparing what we obtain with the standard result in debugging dataset. Specifically, we compare the result with mean square error for both transformed point and

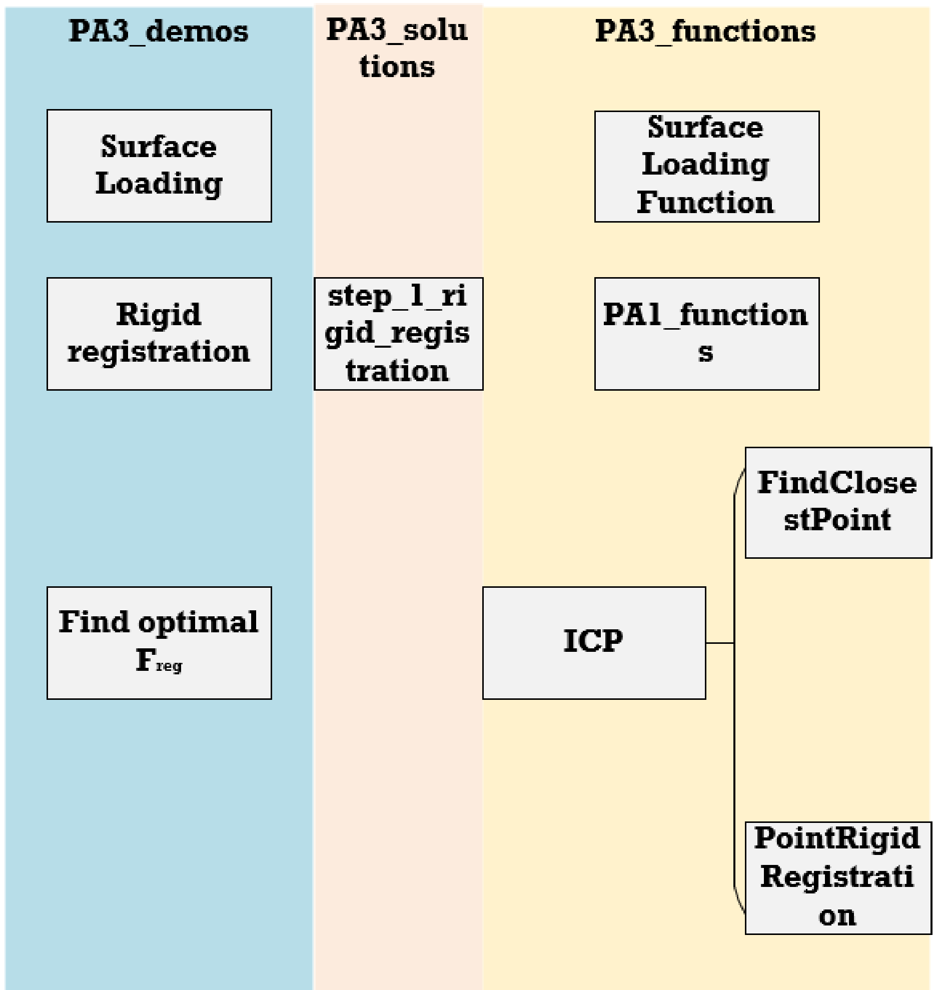

Figure 2: Overview of the program

found closest point on the surface. The result is shown in Table 1 and we can see that the bias between our and standard result is acceptable.

5 Result on unknown data

Result of the output is saved in the output folder and the meaning of each file is given in Table 2.

6 Contribution

Zhiyuan Ding works on the program construction and part of the report writing. Remus Li helps in the debugging process and part of the report writing.

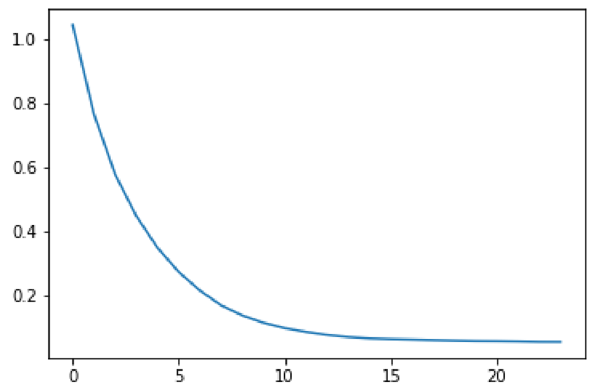

Figure 3: Our Result and given correct result

Table 1: Error of result between our and standard result

data set	error of transformed point	error of closest point
a	0.0066	0.0062
b	0.0153	0.0144
c	0.0218	0.0201
d	0.0134	0.0118
e	0.0256	0.0208
f	0.0338	0.0278

Table 2: Output Tabular for data set

Name	Description
NAME-answer.txt	required output file containing the coordinate and nearest point in surface
NAME-Frame.npy	corresponding frame each ICP obtains