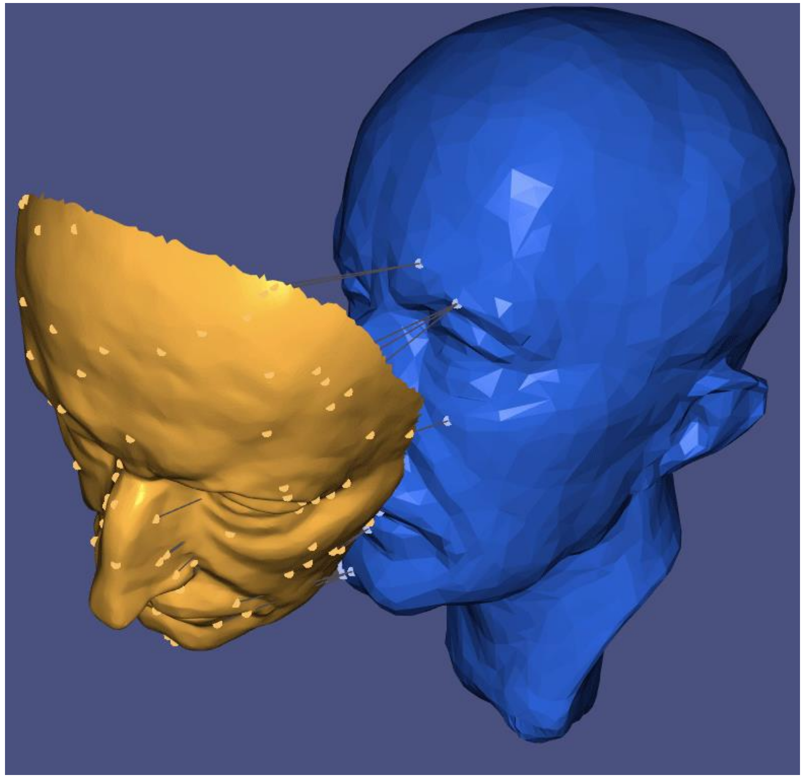
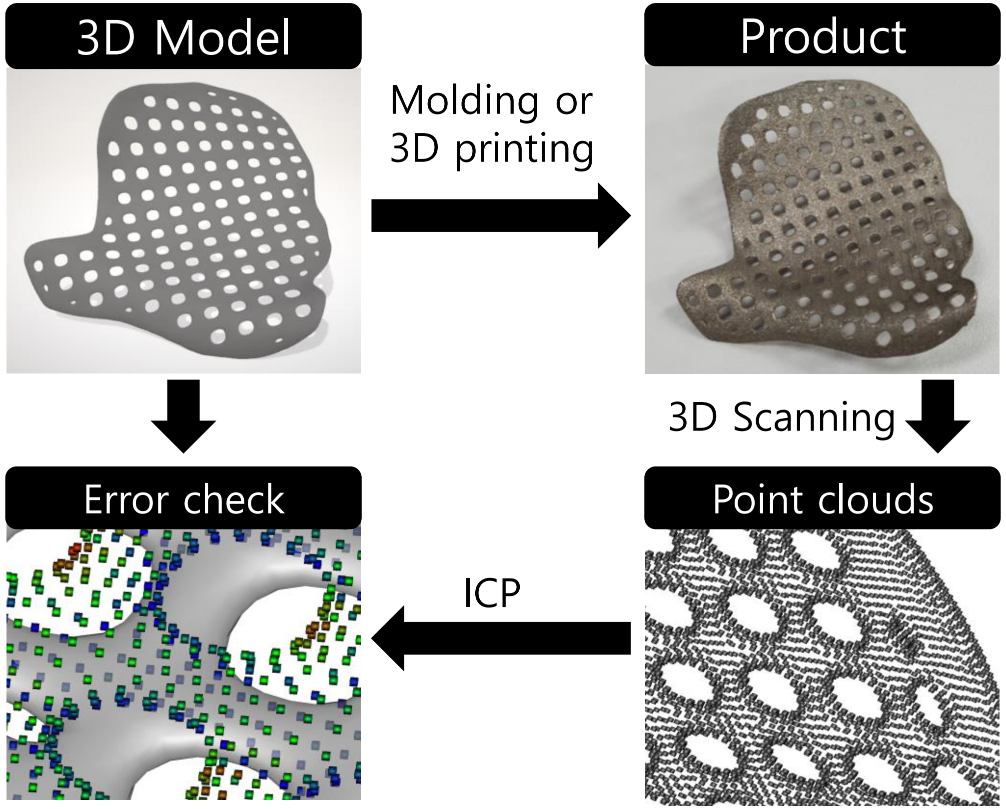
Iterative Closest Point(ICP)
with Genetic Algorithm

UDEOK SEO

1. What is ICP?



< Example of ICP >



< Error check process >

- ICP is an algorithm to minimize the difference between two 3D objects(Mesh, point cloud, etc.) for 3D object fitting.
- ICP is used to measure an error of moldings and 3D-printed objects, or to reverse-engineer.

2. ICP Algorithm

Input : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1, q_2, \cdots, q_m$)

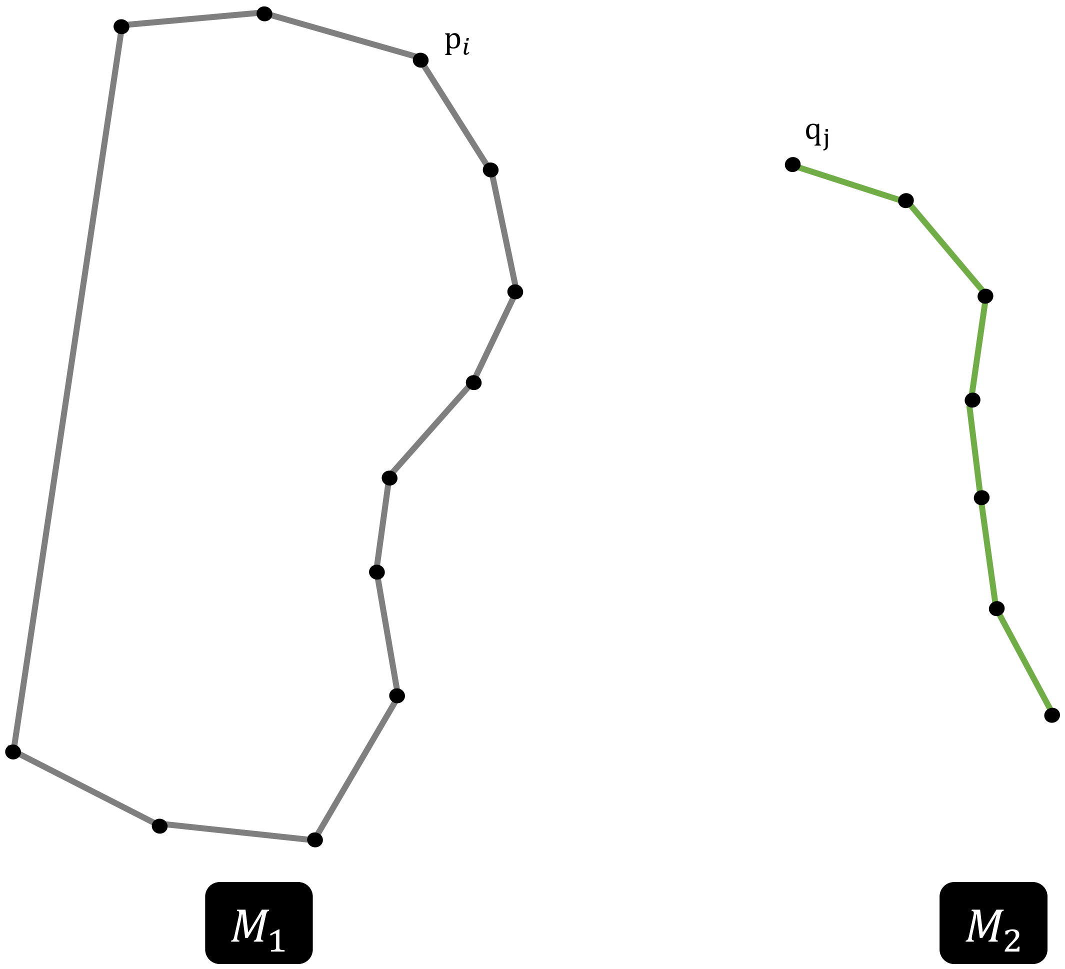
Step1 : Find $\mathrm{q_j}$'s closet points $\mathrm{p_i}$.
$(1 \le \mathrm{i} \le \mathrm{n}, 1 \le \mathrm{j} \le \mathrm{m})$

Step2 : Compute optimal transformation,
$Rotation\ vector\ V_r(\mathrm{r_x}, \mathrm{r_y}, \mathrm{r_z})$,
$Translation\ vector\ V_t(t_x, t_y, t_z)$

Step3 : Update M_2 using V_r, V_t.
$q'_j = Matrix_{rotation} * q_j + V_t \ (1 \le \mathrm{j} \le \mathrm{n})$

Step4 : Check a stopping criteria.
$if\ (\ CheckCriteria()\ )$
$\quad finish\ algorithm$
$else$
$\quad repeat\ Step1 \sim Step3$

Output : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $\mathrm{q}'_1, q'_2, \cdots, q'_m$)



2. ICP Algorithm

Input : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1, q_2, \cdots, q_m$)

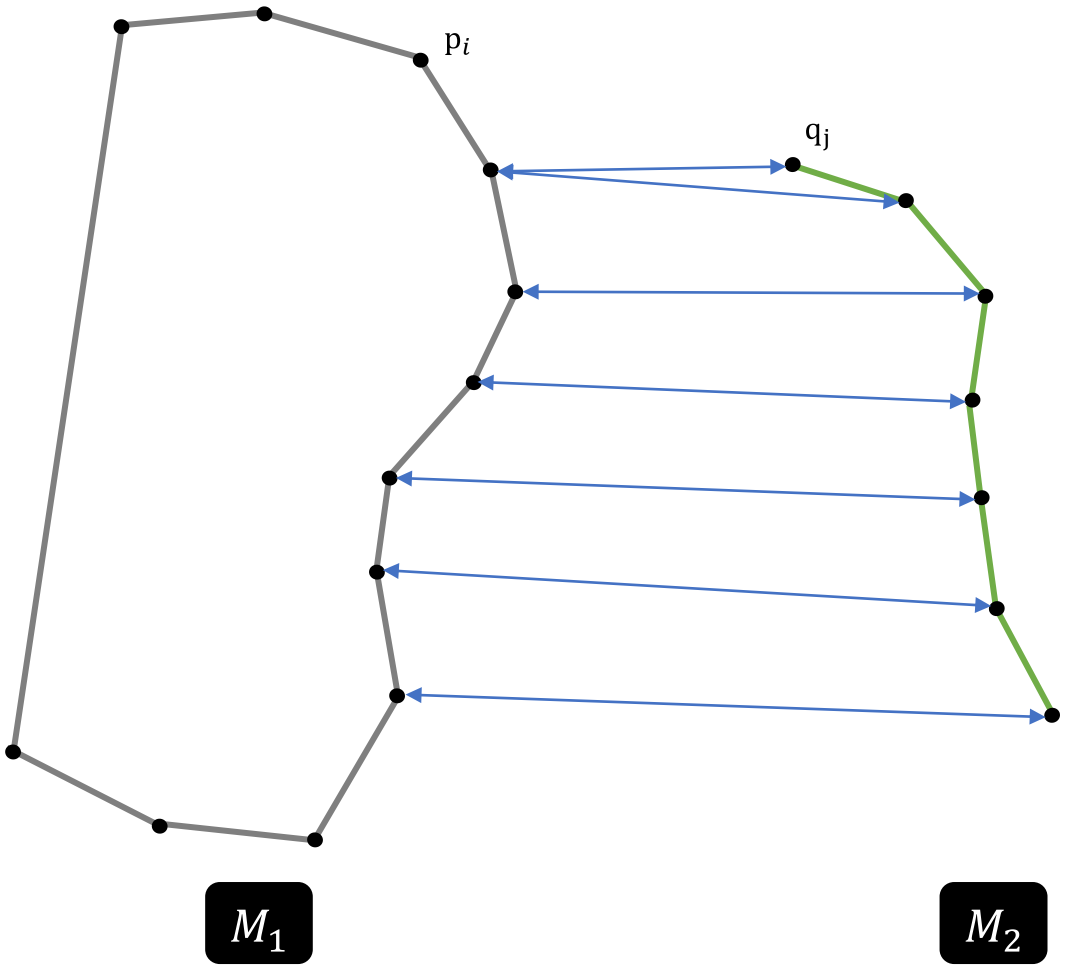
Step1 : Find q_j's closet points p_i.
$(1 \le i \le n, 1 \le j \le m)$

Step2 : Compute optimal transformation,
$Rotation\ vector\ V_r(r_x, r_y, r_z)$,
$Translation\ vector\ V_t(t_x, t_y, t_z)$

Step3 : Update M_2 using V_r, V_t.
$q'_j = Matrix_{rotation} * q_j + V_t \ (1 \le j \le n)$

Step4 : Check a stopping criteria.
$if\ (\ CheckCriteria()\ )$
$finish\ algorithm$
$else$
$repeat\ Step1 \sim Step3$

Output : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q'_1, q'_2, \cdots, q'_m$)



2. ICP Algorithm

Input : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1, q_2, \cdots, q_m$)

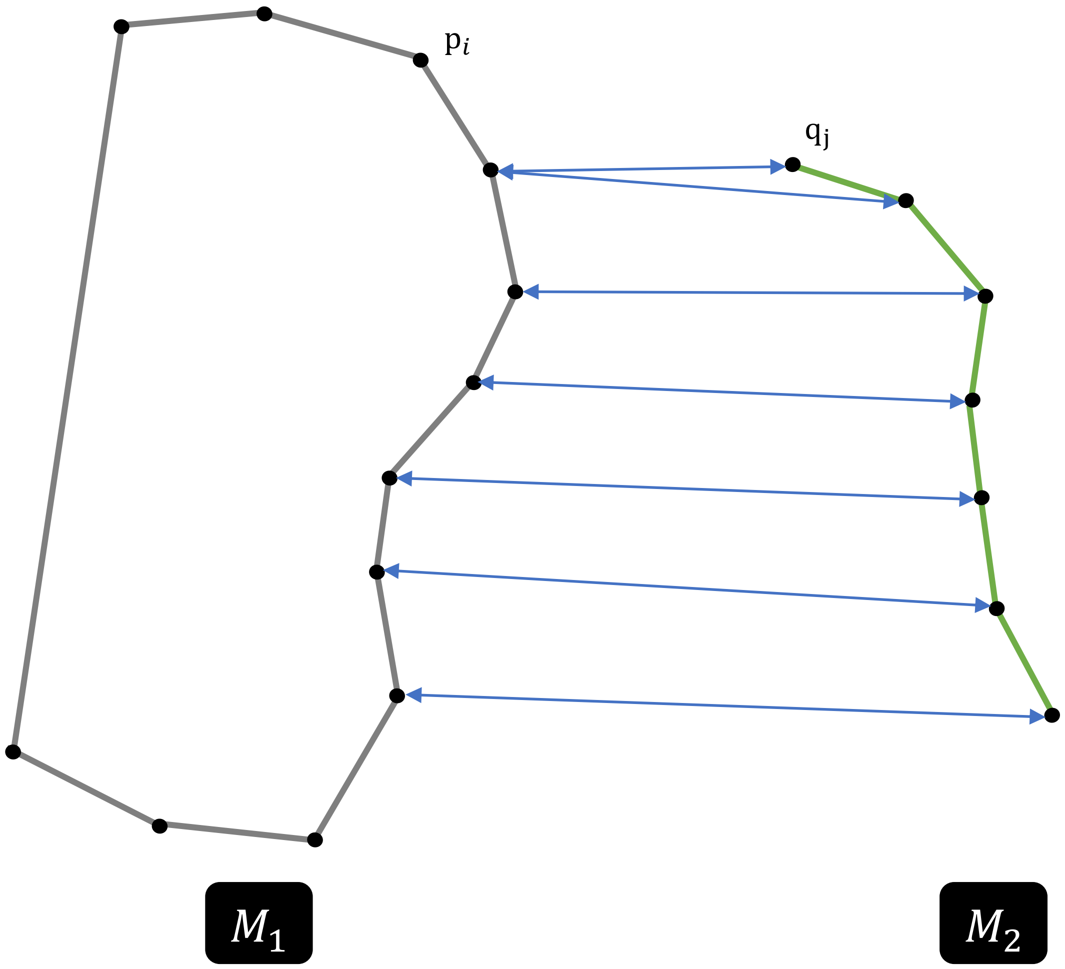
Step1 : Find $\mathrm{q_j}$'s closet points $\mathrm{p_i}$.
$(1 \leq \mathrm{i} \leq \mathrm{n}, 1 \leq \mathrm{j} \leq \mathrm{m})$

Step2 : Compute optimal transformation,
$Rotation\ vector\ V_r(\mathrm{r_x}, \mathrm{r_y}, \mathrm{r_z})$,
$Translation\ vector\ V_t(t_x, t_y, t_z)$

Step3 : Update M_2 using V_r, V_t.
$q'_j = Matrix_{rotation} * q_j + V_t \ (1 \leq \mathrm{j} \leq \mathrm{n})$

Step4 : Check a stopping criteria.
$if\ (\ CheckCriteria()\ )$
$finish\ algorithm$
$else$
$repeat\ Step1 \sim Step3$

Output : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q'_1, q'_2, \cdots, q'_m$)



2. ICP Algorithm

Input : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1, q_2, \cdots, q_m$)

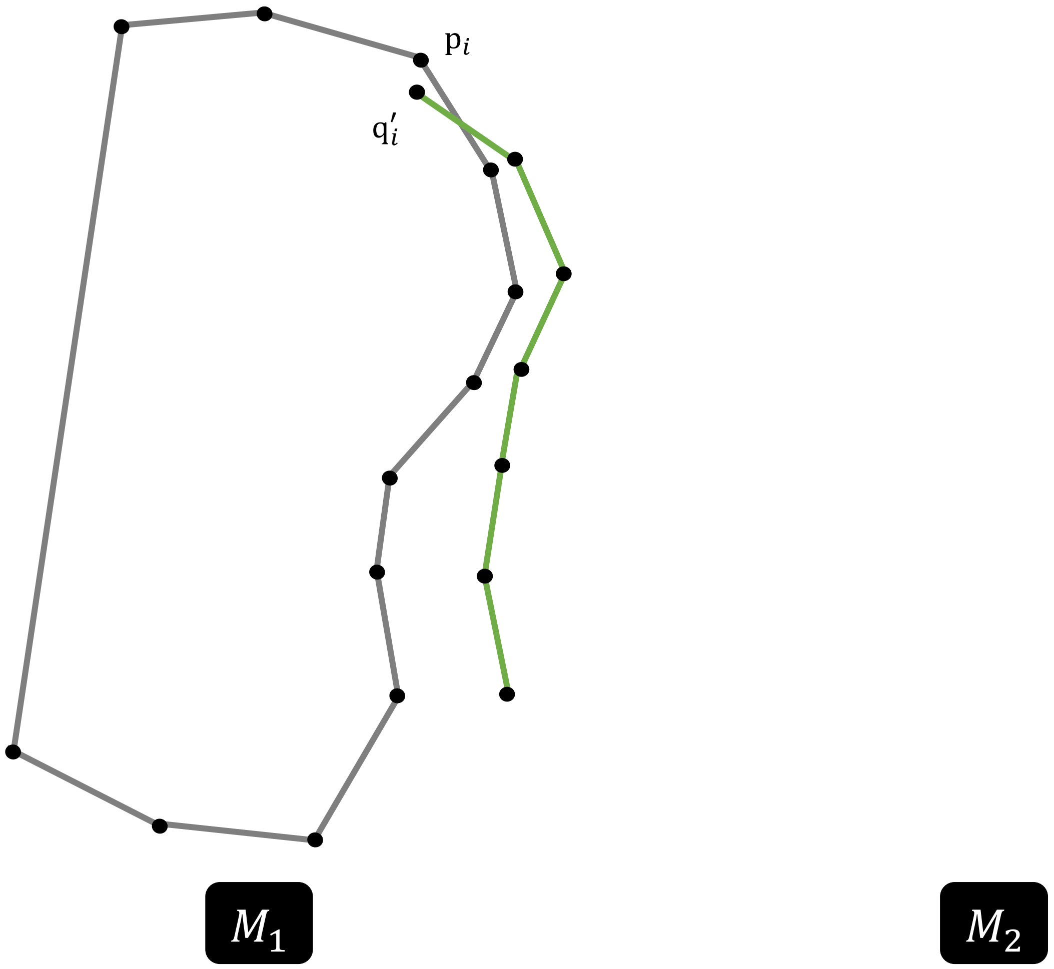
Step1 : Find $\mathrm{q_j}$'s closet points $\mathrm{p_i}$.
$(1 \le \mathrm{i} \le \mathrm{n}, 1 \le \mathrm{j} \le \mathrm{m})$

Step2 : Compute optimal transformation,
$Rotation\ vector\ V_r(\mathrm{r_x}, \mathrm{r_y}, \mathrm{r_z})$,
$Translation\ vector\ V_t(t_x, t_y, t_z)$

Step3 : Update M_2 using V_r, V_t.
$q_j' = Matrix_{rotation} * q_j + V_t \ (1 \le \mathrm{j} \le \mathrm{n})$

Step4 : Check a stopping criteria.
$if\ (\ CheckCriteria()\ )$
$finish\ algorithm$
$else$
$repeat\ Step1 \sim Step3$

Output : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1', q_2', \cdots, q_m'$)



2. ICP Algorithm

Input : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1, q_2, \cdots, q_m$)

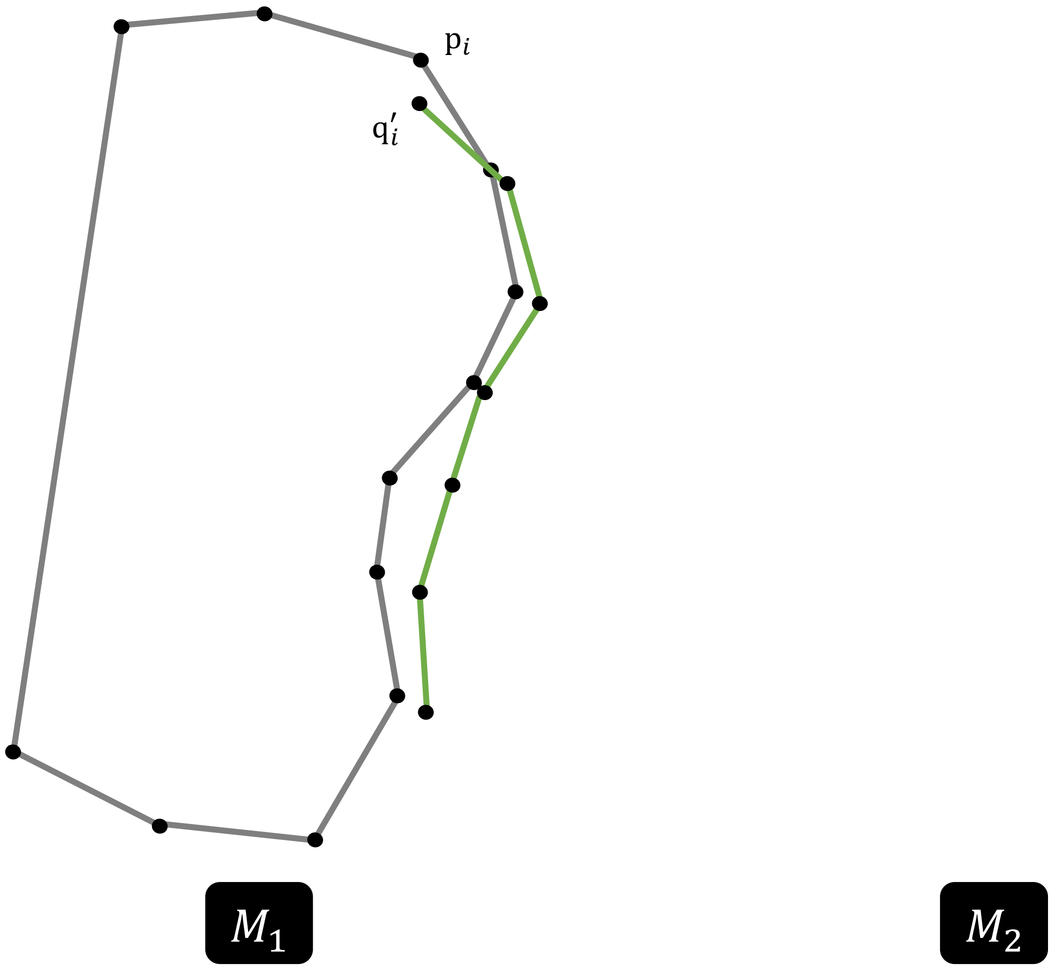
Step1 : Find $\mathrm{q_j}$'s closet points $\mathrm{p_i}$.
$(1 \le \mathrm{i} \le \mathrm{n}, 1 \le \mathrm{j} \le \mathrm{m})$

Step2 : Compute optimal transformation,
$Rotation\ vector\ V_r(\mathrm{r_x}, \mathrm{r_y}, \mathrm{r_z})$,
$Translation\ vector\ V_t(t_x, t_y, t_z)$

Step3 : Update M_2 using V_r, V_t.
$q_j' = Matrix_{rotation} * q_j + V_t$ $(1 \le \mathrm{j} \le \mathrm{n})$

Step4 : Check a stopping criteria.
$if\ (\ CheckCriteria()\ )$
$finish\ algorithm$
$else$
$repeat\ Step1 \sim Step3$

Output : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1', q_2', \cdots, q_m'$)



2. ICP Algorithm

Input : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1, q_2, \cdots, q_m$)

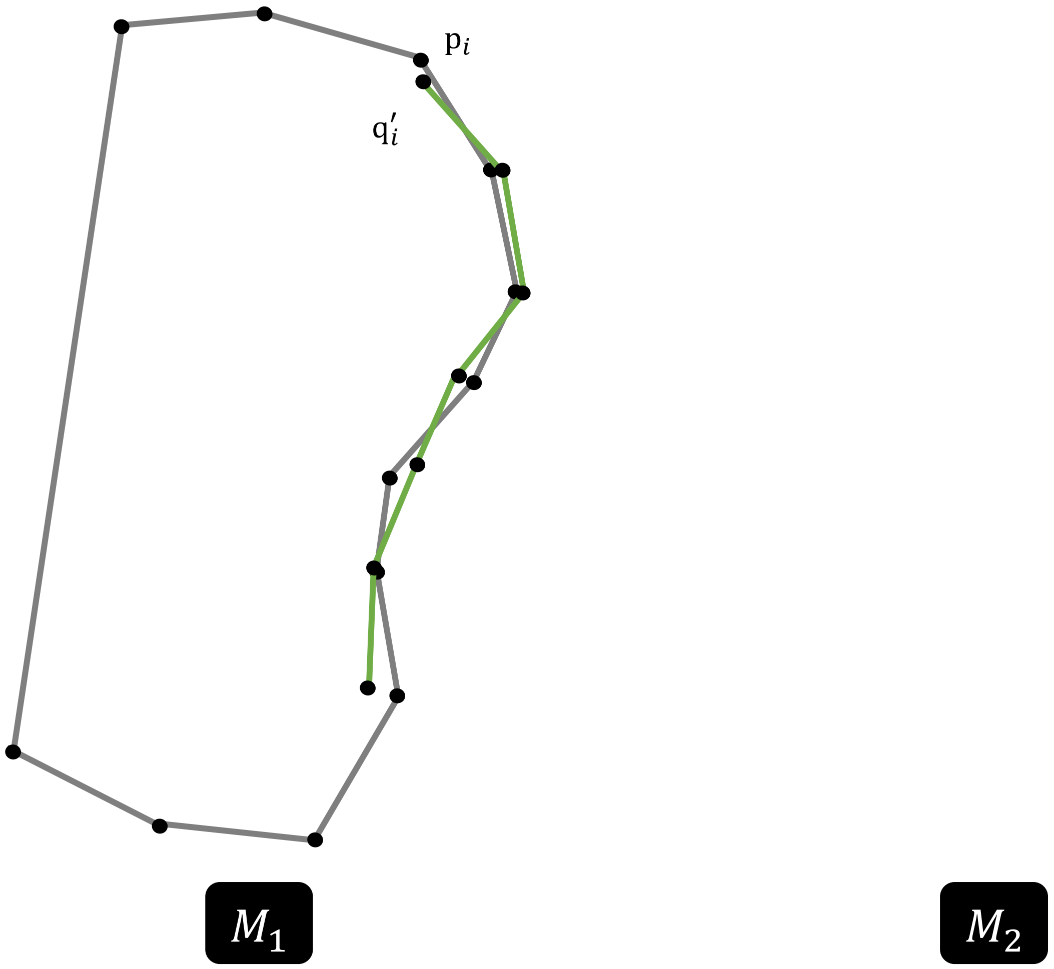
Step1 : Find $\mathrm{q_j}$'s closet points $\mathrm{p_i}$.
$(1 \le \mathrm{i} \le \mathrm{n}, 1 \le \mathrm{j} \le \mathrm{m})$

Step2 : Compute optimal transformation,
$Rotation\ vector\ V_r(\mathrm{r_x}, \mathrm{r_y}, \mathrm{r_z})$,
$Translation\ vector\ V_t(t_x, t_y, t_z)$

Step3 : Update M_2 using V_r, V_t.
$q'_j = Matrix_{rotation} * q_j + V_t$ $(1 \le \mathrm{j} \le \mathrm{n})$

Step4 : Check a stopping criteria.
$if\ (\ CheckCriteria()\ )$
$finish\ algorithm$
$else$
$repeat\ Step1 \sim Step3$

Output : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $\mathrm{q}'_1, q'_2, \cdots, q'_m$)



2. ICP Algorithm

Input : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q_1, q_2, \cdots, q_m$)

Step1 : Find q_j's closet points p_i.
$(1 \le i \le n, 1 \le j \le m)$

Step2 : Compute optimal transformation,
$Rotation\ vector\ V_r(r_x, r_y, r_z)$,
$Translation\ vector\ V_t(t_x, t_y, t_z)$

Step3 : Update M_2 using V_r, V_t.
$q'_j = Matrix_{rotation} * q_j + V_t \quad (1 \le j \le n)$

Step4 : Check a stopping criteria.
if (CheckCriteria())
finish algorithm
else
repeat Step1 ~ Step3

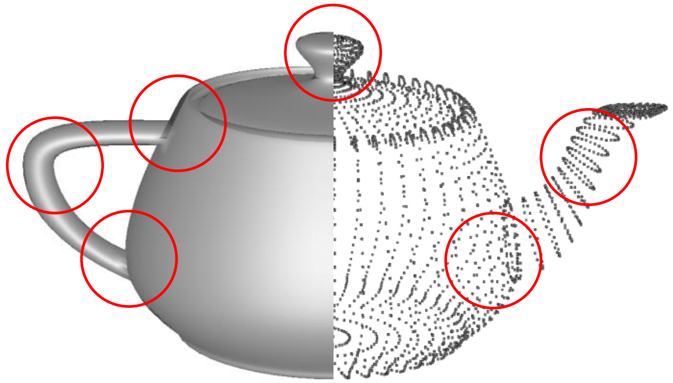
Output : 3D object M_1 (point $p_1, p_2, \cdots, p_n$),
3D object M_2 (point $q'_1, q'_2, \cdots, q'_m$)

Use Genetic Algorithm

for finding two vector, V_r, V_t.

3D object M_1

- Utah teapot
- The number of points is 4,716.



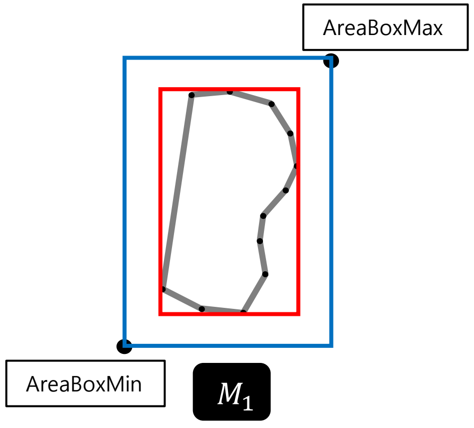
3D object M_2

- Each M_2 is sampling point set of Utah teapot.
- The number of points of M_2 is 170~200.

3. Genetic algorithm for ICP

Individual

- Each individual has 6 value, $\{r_x,\ r_y,\ r_z,\ t_x,\ t_y,\ t_z\}$, *and* $0.0 \le r_x,\ r_y,\ r_z \le 359.9$, $\mathrm{AreaBoxMin.xyz} \le t_x,\ t_y,\ t_z \le \mathrm{AreaBoxMax.xyz}$.
- AreaBox is a clipping box of interest-area, and the size of AreaBox is one and a half times larger than clipping box of 3D object M_1.

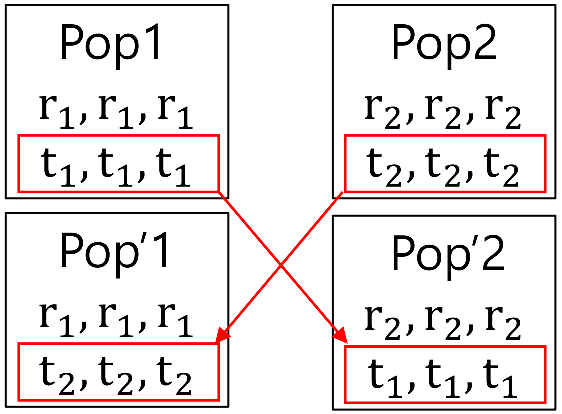


Population

- Population size is 2,000, changed from 1,000 to 2,000.

Crossover

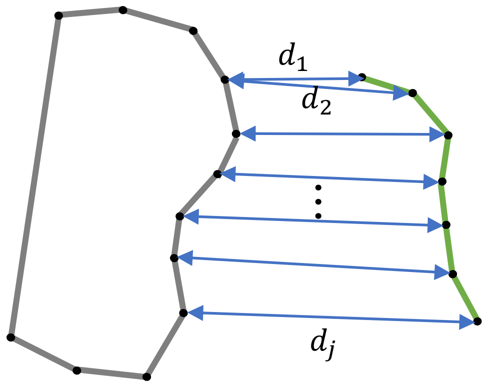
- Selection operator for crossover is 'random selection'.
- Crossover operator is an exchange $\{r_x,\ r_y,\ r_z\}$ or $\{t_x,\ t_y,\ t_z\}$ between two population.



3. Genetic algorithm for ICP

Fitness function

- Fitness function is the function to compute average value of closest point distance.
- Closest point distance is a shortest distance between two points, each points belong to different object(3D mesh or point cloud).



$$\text{Fitness Function} = \frac{\sum_{j=0}^{m} d_j}{m}$$

Mutation

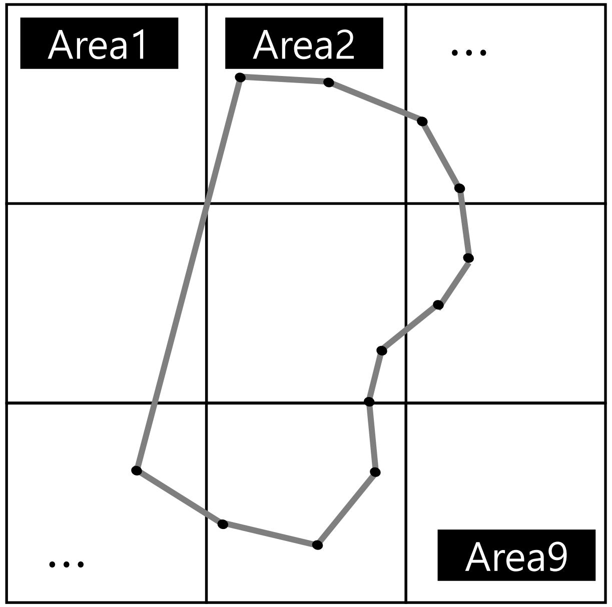
- N best individuals are mutated. (N = 2/10 * POP_SIZE)
- Mutation operator changes each value in an individual to 50%.
  - For N/2 best individuals, mutation adds or subtracts a small random value to the value.
  - For next N/2 best individuals, this function change the value to random value.

Generational selection

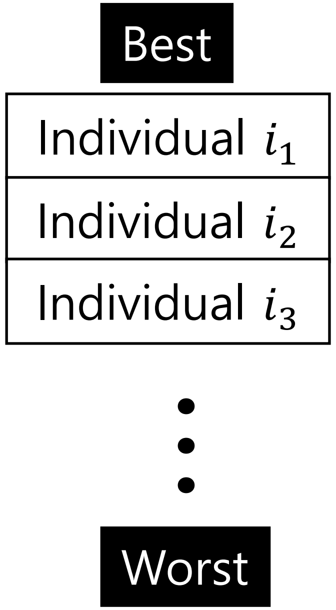
- Elitism, maintain half the individual and make new individuals randomly.

4. Result of Genetic algorithm for ICP

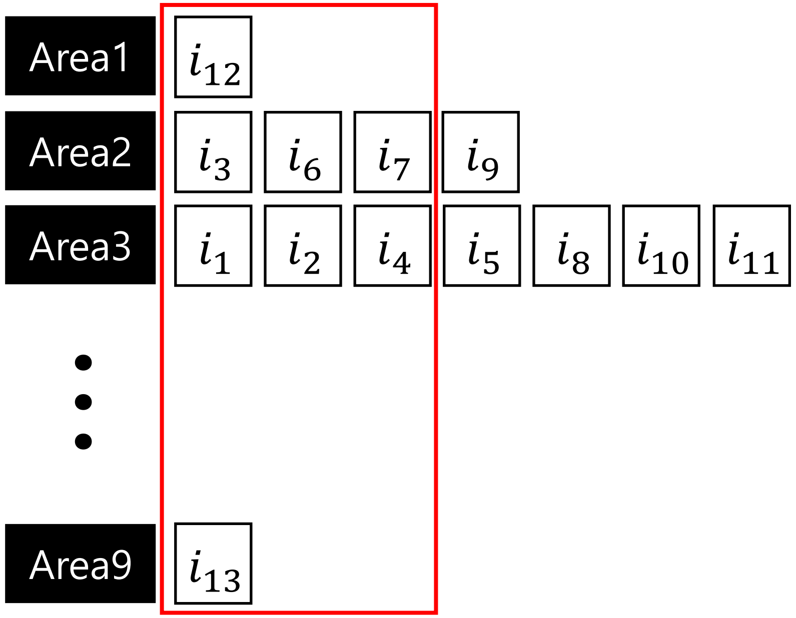
5. How to solve local minimum problem



< Step1 : divide the area >



< Step2 : Sort the nextPop >



< Step3 : Maintain best individual>

- I think there are too many similar individuals in population throughout some generations.
- New elitism : Divide the total area and maintain 100~200 best individual in each partial area.
- After crossover and mutation, add some step for new elitism.

6. Result of Genetic algorithm for ICP

7. Conclusion

❖ ICP with genetic algorithm can find the best answer.

- But this algorithm takes long time and is easy to find a local minimum answer.
- For minimizing the local minimum problem, this algorithm need more time.

❖ If GE is modularized and use 2 or more GE module, the calculation time will be reduced.

- But it is hard to find a good and global threshold for changing GE module.