

고속 전역 정합법에서 정밀도 및 속도 향상을 위한 매개변수 분석

임석현*

Parameter analysis in Fast Global Registration to improve accuracy and speed

Sukhyun Lim*

*Principal Engineer, Innovation Center, 3D Systems Korea, 525 Seolleung-ro, Gangnam-gu, Seoul, 06149 Korea

요 약

정합은 고유 좌표를 가지고 있는 점군을 전역 좌표로 변환하는 과정이다. 지역 정합은 계산 시간이 오래 걸리고 대략적인 위치를 맞춘 후 정밀 정합을 수행하고, 전역 정합은 정합에 이용할 대응점을 계산하고 한 번에 정합하기 때문에 일반적으로 지역 정합법에 비해 속도가 빠르고, 초기 위치에도 상관이 없다. 전역 정합 방법 중 고속 전역 정합법은 성능이 우수하여 많이 사용하는 방법 중 하나이다. 하지만 정합 정밀도와 속도를 높이기 위해서는 많은 매개변수가 필요하다. 본 논문에서는 이와 같은 매개변수들을 분석하고 실험하여 실제 정합 시 유효하게 작용하는 매개변수를 제안한다. 제안한 결과는 고속 전역 정합법을 활용해야 하는 경우 방향 설정에 도움이 될 것이다.

ABSTRACT

The transforming process of point clouds with its local coordinates into a global coordinate is called registration. In contrast to the local registration which takes a long time to calculate and performs precision registration after initial rough positioning, the global registration calculates the corresponding points for registration and performs at once, so it is generally faster than the local registration, and can perform it regardless of the initial position. Among the global methods, the Fast Global Registration is one of the widely used methods due to its fast performance. However, lots of parameters should be set to increase the registration accuracy and speed. In this paper, after analyzing and experimenting the parameters and propose parameters that work effectively in actual registration. The proposed result will be helpful in setting the direction when it is necessary to use the Fast Global Registration method.

키워드 : 정합, ICP, RANSAC, 고속 전역 정합법

Keywords : Registration, ICP, RANSAC, Fast global registration

Received 7 April 2021, Revised 13 April 2021, Accepted 15 April 2021

* Corresponding Author Sukhyun Lim(E-mail:cgwizard@hotmail.com, Tel: +82-2-3218-0510)

Principal Engineer, Innovation Center, 3D Systems Korea, 525 Seolleung-ro, Gangnam-gu, Seoul, 06149 Korea

Open Access <http://doi.org/10.6109/jkiice.2021.25.6.799>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

대상물에 대해 3차원 스캐너로 스캐닝을 한다면 원점이 존재하게 되고, 이 원점을 중심으로 하는 기저 벡터인 지역 좌표계(local coordinate)에 기반을 두어 데이터로 기록이 된다. 일반적으로 스캐닝하는 과정은 스캐너를 움직이면서 스캔하거나 물체를 이동시키면서 각 부분을 스캔하게 되는데, 이들은 스캐너의 지역 좌표계를 기준으로 각 점들이 정의되어 있다. 이들을 공통된 좌표계로 일치시키는 것을 정합(registration)이라고 한다[1, 2].

정합 과정의 정밀도를 높이기 위해서는 스캐너로부터 입력받은 점군 데이터의 잡음(noise)을 적절히 고려해야 한다. 정합 알고리즘이 잡음에 둔감하게 반응하지 않으면 정합 정밀도를 높일 수 없다. 잡음을 효과적으로 처리하기 위하여 사용하는 방법 중 대표적인 것이 RANSAC(RANdom SAMple Consensus)이다[3]. 무작위로 샘플 데이터들을 선택한 다음에 최대 일치(consensus)된 것을 선택하여 대략적인 정렬을 수행한다. 최소자승법(least square method)을 이용할 경우 점들의 거리를 최소화하도록 모델을 찾지만, RANSAC은 일치가 되는 것이 최대(가장 많은 수의 데이터들로부터 지지를 받는 모델)를 선택)로 하는 방법이다.

대략적인 정렬을 하고 나면 미세 조정을 위해 지역 정합(local registration) 알고리즘을 수행하며, 가장 많이 사용하는 대표 방법으로 ICP(Iterative Closest Point)가 있다[4, 5]. ICP는 앞서 RANSAC 등을 이용한 대략적인 정합이 끝난 후에 시작하고, 가장 가까운 점을 선택하여 정합 정밀도 오차가 적어지거나 유저가 정한 최대 횟수까지 반복한다. 이와 같이 대략적인 정합을 하고 지역 정합을 하는 방식은 실제 정합에서 많이 사용되며 다양한 방법이 제안되었다[6]. 하지만 이와 같은 방식은 대략적인 정합을 하는데 있어서 반복적인 작업으로 인한 계산 비용이 많이 들고, 지역 정합에서도 정합 정밀도를 올리기 위해 반복 작업을 해야 하기 때문에 비용이 크다.

앞서 언급한 대략적인 정합을 통해 위치를 잡고 지역 정합을 하는 방식과 달리, 초기 위치 등에 상관없이 한번에 정합하는 방식을 전역 정합(global registration)이라고 한다. 지역 정합 방법이 크게 대략적인 정합 및 미세 조정의 두 단계로 수행되는데 반해 전역 정합은 한 단계로 정합되고, 결과적으로 정합 시간을 단축할 수 있

다. 앞서 지역 정합 방법이 무작위로 샘플 데이터를 추출하는데 반해 전역 정합은 정합을 하고자 하는 두 점군 사이에서 정합에 이용될만한 후보 점들을 구하고 이들로부터 정합을 수행한다.

전역 정합 방법 중 오픈 소스(open source)로 공개되어 많은 사람들이 사용하고 있으며 기존 방법에 비해 정합 속도도 빠르고 정밀도가 높은 방법으로 고속 전역 정합법(Fast Global Registration)이 있다[7, 8]. 하지만 고속 전역 정합법을 이용하여 제대로 된 정합을 수행하기 위해서는 해당 알고리즘에서 사용하는 많은 매개변수 값을 사전에 지정해야 한다. 더군다나 대부분의 변수가 데이터에 의존적이다.

본 논문에서는 고속 전역 정합법을 수행함에 있어서 사용되는 매개변수의 변화에 따른 정합 정밀도를 살펴보고, 유효하게 작용하는 변수를 판단한다. 이를 통해 데이터의 특징에 둔감한 변수 값과 민감하게 반응하는 변수 값을 구분할 수 있고, 결과적으로 정합 정밀도를 올릴 수 있다.

II. 고속 전역 정합법

본 절에서는 전역 정합에서 가장 속도가 빠르고 정합 정밀도가 좋다고 알려져 있는 고속 전역 정합법을 살펴보고 속도와 정합 정밀도 향상을 위하여 필요한 입력 매개변수를 알아본다.

고속 전역 정합법을 수행하기 위해서는 크기 두 단계를 수행한다. 첫 번째로 점군들에서 정합하기 위하여 필요한 정보를 계산하기 위하여 고속 특징점 히스토그램(Fast Point Feature Histogram)을 계산한다[9]. 다음 단계는 이렇게 계산된 히스토그램 정보를 이용하여 실제 정합을 한다. 각 점군에서 계산된 히스토그램을 모두 이용할 경우 속도가 느리고 정확도를 판단하는데 모호성이 증가하기 때문에 고속 정합 정합법에서는 정밀도 및 속도 향상을 위한 입력 매개변수가 필요하다. 이 매개변수에 따라서 정합의 정확도가 향상되거나 정합이 실패할 수도 있으며 속도에도 영향을 미친다.

고속 특징점 히스토그램은 점군에서 각 점마다 사용자가 정한 반경 안의 이웃한 점들과 기하학적 관계를 계산하여 정합을 하게 되고, 이 정합하는 방법 중 가장 효과적으로 알려진 것이 고속 전역 정합법이다.

2.1. 고속 특징점 히스토그램 계산

점 군 P와 Q가 있다고 가정하고 해당 점군의 점들을 p, q , 이들의 고속 특징점 히스토그램 값을 $F(p), F(q)$ 라 가정한다. 이들 사이를 정합하는 강제 변환 행렬을 T 라고 한다.

고속 특징점 히스토그램을 구하기 위해서는 먼저 법선 벡터(normal vector)가 필요하다. 점군에서 점들은 3차원 위치 정보만을 가지고 있는 경우가 대부분이기 때문에 이들 정보만 가지고서는 정합에 필요한 정보를 구하기 어렵다. 점군 데이터는 표면(surface)이 없기 때문에 일반적인 외적 연산(scalar product)으로 법선 벡터를 구할 수 없다. 대신에 이웃 점들을 이용하여 법선 벡터를 구한다. 여기서 고속 전역 정합법을 계산하는데 첫 번째 매개변수가 있는데, 바로 '(i) 법선 벡터를 계산하는데 있어서 참조하는 이웃 점 개수'이다. 이 개수가 많으면 많을수록 전체 형상을 반영하는 법선 벡터를 계산할 수 있지만, 점군에서는 기하학적인 요소가 없기 때문에 인접한 점을 구하기 위해 일반적으로 비용이 많이 들어가는 k-d tree와 같은 자료구조를 이용하기 때문에 속도가 느리다.

다음으로, 임의의 한 점 p_i 와 인접한 점들 간에 고속 특징점 히스토그램을 계산한다. '(ii) 고속 특징점 히스토그램을 계산함에 있어서 인접한 점들을 참조하는 개수 또는 반경'은 사용자가 정하게 되며 이는 고속 전역 정합법을 계산하는데 두 번째 매개변수가 된다. 히스토그램은 p_i 을 중심으로 (ii)에서의 설정한 인접 점들과 1:1 매칭을 통하여 계산하고, 이를 (ii)에서의 모든 점에서 반복하기 때문에 계산량이 많다. 반경을 크게 하거나 최대 참조 포인트 개수가 많으면 이들 간에 히스토그램 계산으로 인하여 정합 정밀도는 올라가지만 반대로 계산 시간이 오래 걸린다.

법선 벡터와 이웃 점 p_j 들의 위치를 나타내는 방향 벡터와의 계산을 통해서 p_i 에서 직교하는 Darboux frame u, v, w 벡터를 도출한다.

식 1과 그림 1을 보면 세 개의 벡터를 확인할 수 있다. 즉, u 는 p_i 의 법선 벡터가 되고, v 는 p_i 에서 p_j 로의 벡터와 u 가 이루는 평면의 법선 벡터이다. w 는 u, v 가 이루는 평면의 법선 벡터이다.

$$u = n_i, v = (p_j - p_i) \times u, w = u \times v \quad (1)$$

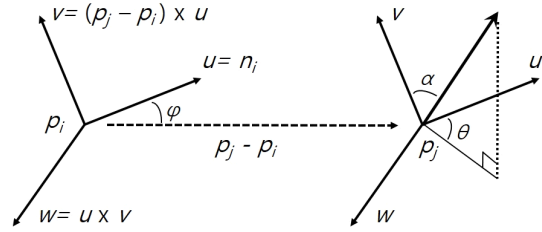


Fig. 1 Fast Point Feature Histogram between p_i and p_j

고속 특징점 히스토그램은 각 점들의 법선 벡터와 w, u, v 와의 관계를 사잇각과 내적을 통하여 α, φ, θ 을 구한다. 이 값들은 서로 다른 벡터와의 스칼라 값들이다.

$$a = v \cdot n_j \quad (2)$$

$$\varphi = (u \cdot (p_j - p_i)) / \|p_j - p_i\| \quad (3)$$

$$\theta = \arctan(w \cdot n_j, u \cdot n_j) \quad (4)$$

α, φ, θ 값들은 전부 범위가 정해져 있기 때문에 중간 값을 계산할 수 있다. α 의 경우 내적 연산이므로 두 벡터가 정규화되어 있다면 삼각함수 \cos 이므로 $-1 \sim +1$ 사이의 값을 가지므로 중간값은 0이다. φ 의 경우도 마찬가지이며, θ 은 \arctan 이므로 $-\pi/2 \sim +\pi/2$ 사이의 값을 가지며 역시 중간값이 0이다. p_i 와 p_j 에서 α, φ, θ 가 구해지고 나면 이 값이 중간값을 기준으로 작은지 큰지를 알 수 있고, 중간값보다 작을 때와 클 때를 각각 0, 1로 나타내게 되면 세 가지 값에 대해 두 가지 경우의 수가 나온다.

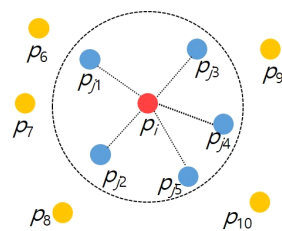


Fig. 2 The influence region for a Fast Point Feature Histogram. The current point (red) and its neighbors (blue) are required. In this figure, the number of neighbors is five

이와 같은 p_i 와 p_j 사이에 쌍(pair)을 맺고 식 2, 식 3, 식 4의 고속 특징점 히스토그램을 계산하고 누적한다. 그림 2와 같이 이후 두 번째 매개변수에서 정의한 다른

이웃인 p_j 와 쌍을 맺고 역시 식 2, 식 3, 식 4 연산을 반복한다. 앞서 언급한 (ii) 매개변수에서 정한 반경 또는 최대 개수만큼 계산한 후 해당 히스토그램을 전부 누적시킨 것을 히스토그램 빈(histogram bin)이라 한다[8]. 이렇게 누적된 히스토그램 빈의 분포는 결국 주변 점과의 각도와 거리가 이루는 상관관계를 나타낸 것이고 이들은 고속 전역 정합에서 사용하는 특징점이 된다.

2.2. 고속 전역 정합법 수행

다음 단계로는 실제로 고속 전역 정합법을 수행하는 단계로서, $F(p)$ 와 $F(q)$ 중에서 실제 정합하기 위한 대응점을 찾는다. 일단 p 의 $F(p)$ 에 대해 대응이 되는 $F(q)$ 를 찾는다 (1차 대응점). 이렇게 찾아진 1차 대응점은 한쪽 방향으로만 고려된 것이기 때문에 q 의 $F(q)$ 에 대해 대응되는 $F(p)$ 를 찾고 이들을 동시에 만족할 경우 2차 대응점으로 간주한다. 여기서 정확성을 높이기 위해 2차 대응점에 대해 랜덤으로 세 점을 택하고 이들의 기하 정보(geometry)을 조사한다. 앞서 2차 대응점만 가지고서는 기하 정보를 고려하지 않고 $F(p)$, $F(q)$ 값만을 이용하여 대응점을 정한 것이기 때문에 모호성이 매우 큰 상태이다. 식 5는 기하 정보를 확인하는 방법이다.

$$\tau < \frac{\|p_i - p_j\|}{\|q_i - q_j\|} < \frac{1}{\tau} \quad (5)$$

세 점간의 상대적인 거리가 사용자가 정한 임계값 τ 범위 안에 들어오는지 확인한다. 이 값이 1.0에 가까울수록 기하 정보가 완벽히 동일할 경우에만 대응점으로 이용한다는 것을 의미한다. 하지만 P와 Q사이에 일반적으로 잡음이 있기 때문에 τ 을 1.0 이하로 세팅하게 되며 그 값이 낮아질수록 정합의 정확도는 낮아지게 된다. 여기서 '(iii) 3차 대응점을 정하기 위한 임계값' τ 가 고속 전역 정합법을 계산하는데 세 번째 매개변수이다.

3차 대응점 테스트까지 만족한 대응점을 $K = \{(p, q)\}$ 라 가정한다. 이 K 를 만족하는 개수가 적으면 정합에 실패할 수 있고, 너무 많으면 모호성이 증가하기 때문에 적절한 값을 사용자가 선택해야 한다. 고속 전역 정합법에 있어서 네 번째 요소는 '(iv) K를 만족하는 대응점의 개수'이다.

다음은 대응점 K 들을 이용하여 변환을 수행하고 오차값(E)가 최소화하는 T 행렬을 구하는 단계이다.

$$E(T) = \sum_{p,q} \rho(\|p - Tq\|) \quad (6)$$

식 6에서 ρ 는 제약 함수(penalty function)이다. 제약 함수는 정답을 정확히 알기 힘든 상황에서 값을 근사적으로 찾기 위해 사용된다. 정합시 잡음 처리를 위해 샘플링을 하고 이들의 유효성을 검사하고 최적화하는 단계를 반복해야 하는데 이는 속도 저하 요소가 된다. 제약 함수 ρ 를 설정하게 되면 정합의 정확한 답을 알 수 없는 상황에서 근사(approximate) 요소로서 작용하여 강건한(robust)한 답을 찾는데 도움이 된다. 제약 함수는 Charbonnier, Cauchy, Welsch 등 많은 방법이 존재하며, 고속 전역 정합법에서는 Geman-McClure 함수를 사용한다[10]. 식 7에서 μ 에 따라 결국 제약 함수 ρ 가 만들어지고, 이 값이 작아지면 작아질수록 정합의 정밀도가 올라간다.

$$\rho(x) = \frac{\mu x^2}{\mu + x^2} \quad (7)$$

식 7의 적절한 μ 값을 알 수 없기 때문에 일반적으로 매우 큰 값을 초기값으로하고, 이 값을 단계적으로 줄여가면서 식 6의 오차값을 최소화하는 방향으로 진행한다. 결과적으로 μ 값을 선택하는 것이 정합의 정밀도와 밀접한 관계가 있으며, 다섯 번째 요소는 '(v) μ 값을 줄여가면서 정합 정밀도를 높여가는 반복 횟수'이다. 물론 사용자가 정한 값에 수렴할 경우 이 반복 횟수보다 적더라도 정합 과정을 끝낸다. 일반적으로는 초반에는 정합의 정밀도가 높아지지만 반복할수록 차이가 적게 되므로 해당 반복 횟수는 빠른 정합 수행을 위해서 필수적이다. 하지만 이 값을 너무 작게 설정할 경우 정합의 정밀도는 낮아지게 된다.

고속 전역 정합법을 수행함에 있어서 여섯 번째 요소는 μ 값을 계속 줄여가면서 반복을 하다가 너무 작을 경우에는 μ 값 줄이는 과정을 멈추어야 한다. 즉 '(vi) μ 값의 최솟값'이다. μ 값이 작아질수록 정합의 정밀도가 증가하기 때문에 최솟값의 설정이 필요하다.

```

Compute normals  $n_p$  and  $n_q$  by referring neighbors; (i)
Compute FPFH  $F(P)$  and  $F(Q)$  by referring neighbors; (ii)
Build corresponding points( $K$ ) (iv) using threshold  $\tau$ ; (iii)

for ( all (iv) correspond points, until iteration count < (iv) )
{
    for ( $p, q \in K$ )
        solving Jacobian;

    compute  $T$  matrix, and update ( $T \times T_{previous}$ );

    if ( $\mu > (vi)$ )
         $\mu = \mu / (vii)$ ;
}
    
```

Fig. 3 The registration pseudo code using Fast Global Registration

마지막으로 일곱 번째 요소는 ‘(vii) μ 값을 줄여나가는 비율’이다. 이 값을 1.0보다는 커야 하며 이 값이 클수록 μ 값이 빠르게 줄어들기는 하지만 자칫 정합의 정밀도를 낮출 수 있기 때문에 적절한 값이 필요하다. 반대로 너무 낮을 경우에는 정합을 위한 반복 횟수가 증가하게 된다.

그림 3은 지금까지 살펴본 요소 일곱 가지를 모두 반영하여 고속 전역 정합법을 이용하여 정합 수행하는 의사 코드(pseudo code)이다.

III. 실험 결과

이 장에서는 고속 전역 정합법을 수행함에 있어서 매개변수 일곱 가지 변화에 따른 정합 정밀도와 속도와의 관계를 비교해본다. 고속 전역 정합법은 공개 코드이고 Open3D™에 포함되어 있기 때문에 실험 환경은 공개된 환경을 그대로 사용한다. 실험에 사용한 데이터는 정합에 널리 사용하는 stanford bunny, buddha, armadillo 모델을 이용한다[11]. 공개된 모델을 그대로 사용할 경우 점군을 구성하는 점 개수가 작아서 매개변수 변화에 따른 정합의 정확도를 판단하기 힘들기 때문에 임의로 추가점을 삽입하여 사용하였다.

매개변수를 변화량에 따른 정합 수행 시, 정밀도를 판단하기 위하여 일반적으로 많이 사용하는 RMSE(Root Mean Square Error)을 사용한다. RMSE는 대응점들 사이에 유클리드언 거리(Euclidean distance)를 제공하고 이를 모두 더한 후 점 개수만큼 나누어 제공근한 것이다.

실험을 위하여 일곱 가지 매개변수의 값들의 기본값을 중심으로 최소 범위와 최대 범위를 잡고 정합 시간과 정합 정밀도를 살펴본다. 실험을 통하여 매개변수 중 일부는 범위를 크게 잡은 것이 있으며, 변화량이 미미하거나 수렴할 경우 상대적으로 작게 잡았다. 각 매개 변수를 실험할 때 나머지 매개 변수는 모두 기본값으로 설정한다. 각각의 매개변수 변화에 따른 모든 실험값을 논문문에 표기하기 힘들기 때문에 최솟값, 기본값, 최댓값 및 중간에 의미 있는 변화가 있는 값을 정리한다. 실험에 이용한 세 가지 데이터의 경향이 비슷했기 때문에 표는 armadillo 모델에 대해서만 정리한다. 처리 시간 단위는 초이고, RMSE 단위는 mm이다.

표 1은 첫 번째 매개변수인 법선 벡터를 계산하는데 있어서 참조하는 이웃 점의 개수에 따른 변화이다. 결과를 보면, RMSE가 1과 같이 의미 없는 낮은 숫자에서 높다가 값이 증가하면서 줄어드는 것처럼 보이지만 경향과 상관없음을 알 수 있다. 예를 들어 15인 경우에는 RMSE 값이 가장 낮으며, 바로 다음 17은 오히려 가장 높은 수치를 보이고 있다. Bunny와 buddha 모델 역시 값이 일관성을 가지지 않고 변하는 것을 확인할 수 있다. 이는 첫 번째 매개변수는 데이터에 의존적이기 때문에 데이터에 맞게 사용자가 설정해야 함을 의미한다.

표 2의 두 번째 매개변수인 고속 특징점 히스토그램을 계산하는데 있어서 참조하는 이웃 점의 개수에 따른 변화를 보면, 첫 번째 매개변수와 비슷하게 일관성을 보기 힘들다. 45, 60, 125일 때 RMSE가 낮은 반면에, 25, 35, 40일 때 RMSE값이 상대적으로 높다. 기본값인 100일 때도 RMSE가 가장 낮다고 볼 수 없다. Bunny와 buddha 모델을 실험하면 역시 값이 일관성을 가지지 않고 변하는 것을 확인할 수 있다. 결국 두 번째 매개변수 역시 데이터에 의존적이기 때문에 데이터에 맞게 사용자가 설정해야 함을 의미한다.

표3의 세 번째 매개변수인 대응점 선별을 위한 임계값은, 최솟값에서부터 기본값인 0.95까지 RMSE가 증가하다가 0.95부터 반대로 높아지고 있다. 즉, 값이 낮으면 선별된 대응점이 정합에 적합하지 않게 선택되어서 정밀도가 낮아졌으며, 높으면 모호성 증가로 오히려 정밀도가 낮아짐을 알 수 있다. 결국 해당 매개변수는 기본값으로 설정하면 데이터에 상관없이 정합 결과를 기대할 수 있다. Bunny와 buddha 모델에서도 동일한 결과를 얻었다.

Table. 1 Processing time and RMSE according to the 1st parameter (default: 30)

value	time	RMSE	value	time	RMSE	value	time	RMSE	value	time	RMSE
1	0.081	1.931	3	0.096	1.728	5	0.092	1.827	7	0.094	1.382
9	0.106	1.248	11	0.107	1.333	13	0.093	1.525	15	0.112	1.142
17	0.088	1.754	19	0.095	1.270	21	0.097	1.335	23	0.100	1.356
25	0.101	1.287	27	0.108	1.445	29	0.107	1.446	31	0.145	1.398
33	0.106	1.330	35	0.122	1.331	37	0.149	1.354	39	0.137	1.357
41	0.127	1.532	43	0.138	1.383	45	0.116	1.405	47	0.177	1.428
49	0.225	1.290	51	0.124	1.495	53	0.149	1.491	55	0.126	1.461
57	0.115	1.441	58	0.148	1.402	59	0.124	1.567	60	0.111	1.538

Table. 2 Processing time and RMSE according to the 2nd parameter (default: 100)

value	time	RMSE	value	time	RMSE	value	time	RMSE	value	time	RMSE
1	0.105	0.000	5	0.048	2.045	10	0.052	1.777	15	0.054	1.641
20	0.051	1.487	25	0.053	1.805	30	0.055	1.878	35	0.055	1.365
40	0.057	1.822	45	0.080	1.246	50	0.075	1.423	55	0.074	1.265
60	0.075	1.221	65	0.080	1.455	70	0.087	1.446	75	0.081	1.444
80	0.093	1.429	85	0.136	1.466	90	0.116	1.499	95	0.131	1.386
100	0.112	1.392	105	0.120	1.355	110	0.148	1.338	115	0.123	1.363
120	0.125	1.412	125	0.177	1.233	130	0.159	1.313	135	0.152	1.318
140	0.178	1.466	145	0.175	1.407	150	0.165	1.285	155	0.148	1.334

Table. 3 Processing time and RMSE according to the 3rd parameter (default: 0.95)

value	time	RMSE	value	time	RMSE	value	time	RMSE	value	time	RMSE
0.70	0.476	1.613	0.72	0.801	1.584	0.74	0.748	1.580	0.76	0.471	1.578
0.78	0.447	1.565	0.80	0.461	1.561	0.82	0.477	1.554	0.84	0.467	1.580
0.86	0.681	1.579	0.88	0.367	1.563	0.90	0.281	1.508	0.92	0.280	1.488
0.94	0.171	1.441	0.95	0.098	1.365	0.98	0.082	1.552	0.99	0.080	0.000

Table. 4 Processing time and RMSE according to the 4th parameter (default: 1000)

value	time	RMSE	value	time	RMSE	value	time	RMSE	value	time	RMSE
10	0.077	1.516	15	0.074	1.466	20	0.076	1.449	25	0.084	1.415
30	0.080	1.408	35	0.094	1.382	40	0.113	1.307	45	0.131	1.315
50	0.102	1.345	55	0.091	1.353	60	0.091	1.334	65	0.114	1.327
70	0.122	1.323	75	0.119	1.352	80	0.130	1.369	85	0.110	1.370
90	0.128	1.363	100	0.112	1.392	200	0.108	1.392	1000	0.108	1.392

Table. 5 Processing time and RMSE according to the 5th parameter (default: 64)

value	time	RMSE	value	time	RMSE	value	time	RMSE	value	time	RMSE
1	0.076	2.126	7	0.073	1.558	13	0.086	1.563	19	0.080	1.583
25	0.095	1.587	31	0.087	1.585	37	0.096	1.565	43	0.098	1.512
49	0.120	1.469	53	0.123	1.450	57	0.124	1.424	60	0.109	1.403
64	0.112	1.392	70	0.146	1.376	77	0.159	1.372	83	0.162	1.352
95	0.172	1.317	105	0.174	1.317	115	0.182	1.317	125	0.198	1.317

Table. 6 Processing time and RMSE according to the 6th parameter (default: 0.025)

value	time	RMSE	value	time	RMSE	value	time	RMSE	value	time	RMSE
0.001	0.110	1.392	0.025	0.118	1.392	0.050	0.113	1.392	0.100	0.110	1.392
0.150	0.107	1.392	0.200	0.131	1.392	0.250	0.112	1.392	0.300	0.112	1.392
0.350	0.110	1.392	0.400	0.124	1.403	0.450	0.129	1.403	0.500	0.112	1.403
0.550	0.112	1.403	0.600	0.110	1.424	0.650	0.126	1.424	0.700	0.113	1.424
0.750	0.120	1.424	0.800	0.109	1.450	0.850	0.112	1.450	0.900	0.117	1.450

Table. 7 Processing time and RMSE according to the 7th parameter (default: 1.4)

value	time	RMSE	value	time	RMSE	value	time	RMSE	value	time	RMSE
1.01	0.143	1.568	1.05	0.108	1.560	1.10	0.152	1.569	1.15	0.112	1.585
1.20	0.111	1.566	1.25	0.117	1.525	1.30	0.118	1.464	1.35	0.137	1.417
1.40	0.110	1.392	1.45	0.125	1.379	1.50	0.148	1.368	1.55	0.117	1.353
1.60	0.118	1.329	1.65	0.112	1.319	1.70	0.144	1.320	1.80	0.139	1.314
1.85	0.109	1.420	1.90	0.115	1.412	1.95	0.113	1.519	2.00	0.113	1.553

표 4의 네 번째 매개변수 역시 값이 증가함에 따라 RMSE값이 낮아지고 있다. 하지만 100이 넘어가면 대응점의 개수 부족으로 수렴하는 것을 볼 수 있다. 처리 시간도 100, 200, 1000일 때 동일하기 때문에 값을 충분히 크게 잡더라도 필요한 개수만큼만 대응점으로 선택된다는 것을 확인할 수 있다. 결국 기본값으로 설정해도 무방하며, bunny와 buddha 모델에서도 동일했다.

표 5의 μ 값을 줄여가는 반복 횟수 변화량에 결과를 보면 기본값에 도달할 때까지 값이 줄어들고 있지만 기본값인 64부터는 그 감소세가 둔해지고 있다. 반면에 시간 역시 길어지고 있다. 즉, 기본값인 64일 때가 수행 시간 대비 낮은 RMSE 값을 보인다고 할 수 있다. 이는 bunny와 buddha 실험에서도 비슷한 경향을 보였다.

표 6의 μ 값을 줄일 수 있는 최댓값의 변화량을 보면 0.35일 때까지 1.392로 동일하다가, 상승하다가 다시 동일하기 진행되는 계단 형태로 RMSE가 증가하고 있다. 기본값인 0.025일 때가 가장 낮기 때문에 여섯 번째 매

개변수도 역시 기본값으로 설정하면 되며 역시 다른 모델에서도 비슷한 경향을 보였다.

표 7의 마지막 μ 값을 줄이는 비율에 따른 변화량을 보면 값을 크게 줄수록 μ 값이 빨리 줄어들어 정합의 정밀도가 높아지지만, 1.8을 기점으로서는 오히려 너무 빨리 줄어들어 정합 정밀도가 가장 높은 위치를 고려하지 않고 있음을 확인할 수 있다. bunny 모델인 경우 armadillo와 비슷했지만, buddha 모델인 경우 기본값인 1.4보다 클 때 RMSE 값이 소폭 증가하였다.

결론적으로 고속 전역 정합법을 이용하여 전역 정합을 할 때 첫 번째, 두 번째 매개변수인 경우 변화량이 크기 때문에 데이터에 맞게 실험에 의존하여 값을 정해야 하고, 일곱 번째 매개변수 역시 기본값을 중심으로 약간씩 증가시키면 정합의 정밀도가 올라가는 것을 확인할 수 있다. 나머지 네 개의 매개변수는 모두 기본값으로 설정하면 되었다.

IV. 결 론

3차원 스캐닝을 이용한 기술이 발전함에 따라 서로 다른 지역 좌표계에서 취득된 점군데이터를 공통된 좌표인 전역 좌표계로 배치시키는 정합 단계의 정밀도의 중요성은 날로 커지고 있다. 대략적인 위치를 맞추고 세부적인 정합을 하는 지역 정합 기법과 달리 전역 정합 기법은 한 번의 수행으로 원하는 결과를 얻을 수 있어 많이 사용되고 있다. 고속 전역 정합법은 지역 정합 기법 포함 다른 전역 정합 기법에 비해서도 성능이 우수하여 많이 사용되고 있지만 제대로 된 정합하기 위해서는 일곱 개의 매개변수를 설정해야만 했다. 본 논문에서는 유효하게 작용하는 법선벡터를 계산하기 위한 이웃 점의 개수, 고속 특징점 히스토그램 계산을 위한 인접 점의 개수 및 제약 함수의 감소 비율의 세 가지 유효한 매개변수를 검증하였다.

추후 연구과제로서는 각각의 매개변수들이 복합적으로 작용했을 때 결과를 검증할 필요가 있으며, 데이터의 의존적인 매개변수들을 자동으로 설정하는 방법에 대한 연구가 필요할 것이다.

REFERENCES

[1] S. Lim, "Effective criterion for evaluating registration accuracy," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 25, no. 5, pp. 652-658, 2021.

[2] K. Kwon, "A weighted points registration method to analyze dimensional errors occurring during shipbuilding process," *Transactions of the Society of CAD/CAM Engineers*, vol. 21, no. 2, pp. 151-158, 2016.

[3] Z. H. Nejad and M. Nasri, "An adaptive image registration method based on SIFT features and RANSAC transform," *Computers & Electrical Engineering*, vol. 62, pp. 524-537, 2017.

[4] M. He, L. Huang, B. Zhao, B. Chen, and B. Hu, "Advanced functional materials in solid phase extraction for ICP-MS determination of trace elements and their species - A review," *Analytica Chimica Acta*, vol. 973, no. 22 pp. 1-24, 2017.

[5] Z. Wu, H. Chen, S. Du, M. Fu, N. Zhou, and N. Zheng, "Correntropy based scale ICP algorithm for robust point set registration," *Pattern Recognition*, vol. 93, pp. 14-24, 2019.

[6] J. Yang, H. Li, D. Campbell, and Y. Jia, "Go-ICP: A globally optimal solution to 3D ICP pointset registration," in *Proceedings IEEE/CVF International Conference on Computer Vision*, 2016.

[7] Q. Y. Zhou, P. Jaesik, and K. Vladlen, "Fast global registration," in *Proceedings European Conference on Computer Vision*, Netherlands, 2016.

[8] Open3D project [Internet]. Available: <http://www.open3d.org/>.

[9] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proceedings IEEE International Conference on Robotics and Automation*, 2009.

[10] J. T. Barron, "A general and adaptive robust loss function," in *Proceeding of Computer Vision and Pattern Recognition*, USA, pp. 4331-4339, 2019.

[11] Stanford University 3D Scan Repository [Internet]. Available: <http://graphics.stanford.edu/data/>.



임석현(Sukhyun Lim)

1999년: 인하대학교 전자계산공학과 공학사
 1999년: 인하대학교 물리학과 이학사
 2001년: 인하대학교 전자계산공학과 공학석사
 2006년: 인하대학교 컴퓨터정보공학과 공학박사
 2009년: 인하대학교 컴퓨터정보공학과 연구교수
 2012년: 한국전자통신연구원 선임연구원
 2018년: NHN NEXT 교수
 2018년 - 현재: 3D systems Korea 수석연구원
 ※관심분야: 컴퓨터그래픽스, 컴퓨터비전, 게임, 통신시스템