

통계 기법을 이용한 곡면 복원

윤민철¹, 이윤진¹, 이승용¹, Ioannis Ivrissimtzis², Hans-Peter Seidel²

포항공과대학교¹, Max Plank Institut Informatik (MPI Informatik)²

Surface Reconstruction Using Statistical Techniques

Mincheol Yoon¹, Yunjin Lee¹, Seungyong Lee¹, Ioannis Ivrissimtzis², and Hans-Peter Seidel²

Pohang University of Science and Technology(POSTECH)¹, Max Plank Institut Informatik (MPI Informatik)²

요약

곡면 복원이나 곡면 복원과 깊은 관련이 있는 노말 추정을 하는 대부분의 방법은 결정론적인 알고리즘을 사용한다. 결정론적 알고리즘은 속도가 빠르고, 오차가 크지 않은 입력에 대해서는 좋은 질의 곡면 복원을 할 수 있다. 그러나 결정론적 방법의 특성상 이상치나 노이즈를 가진 데이터에 대해서는 안정된 복원 결과를 얻을 수 없다. 본 논문에서는 앙상블이라고 불리는 통계적인 방법을 사용해서 곡면 복원과 노말 추정을 하는 기존의 알고리즘을 개선한다. 앙상블 기법은 먼저 입력 점 집합을 무작위로 샘플링해서 점 집합의 부분집합을 만든다. 그리고 나서 만들어진 부분 집합에 독립적으로 결정론적인 알고리즘을 적용하여 결과를 얻어낸다. 마지막으로, 각각의 서로 다른 결과를 결합하여 더 우수한 최종결과를 얻어낸다. 널리 쓰이는 노말 추정 기법[11]과 Multi-level Partitions of Unity implicit [18]를 사용해서 앙상블이 효과적으로 노이즈가 많은 데이터를 처리할 수 있는 것을 보여준다.

1. 서론

삼차원 모델을 만들기 위해서는 많은 시간과 복잡한 과정이 필요하다. 최근 기술이 발전했음에도 불구하고, 삼차원 물체를 만드는 일은 수작업을 필요로 하는 등 비효율적인 면이 많다[4] [22]. 이러한 이유에서 삼차원 콘텐츠는 온라인 쇼핑 같은 중요한 응용분야에서 여전히 널리 사용되지 못하고 있다.

삼차원 모델링의 과정은 먼저 획득 과정(acquisition stage)부터 시작한다. 이 작업에서는 실제 물체를 스캔(scan)함으로써 주로 점 집합(point set)의 형태로 물체의 기하적인 정보(geometric information)를 얻어낸다. 다음 작업은 스캔된 물체의 표면을 만들어 내기 위해 기하 정보를 처리하는 작업이다. 이 작업은 곡면 복원(surface reconstruction)이라고 불린다. 본 논문에서는 이와 같은 곡면 복원 물체를 다룬다.

곡면 복원은 점 집합의 노말 벡터(normal vector)를 구하는 문제와 밀접한 관련이 있다. 사실 좀 더 빠르고, 안정된(robust) 곡면 복원을 하기 위해서는 점 뿐만 아니라 점의 노말 벡터도 필요한 경우가 많다. 이러한 이유에서 노말 벡터를 구하는 일은 곡면 복원을 위해 꼭 필요한 과정이다. 본 논문의 실험에서 밝혀진 바와 같이 기존의 개발된 가장 발전된 기술을 사용한다고 해도 이상치(outlier)나 노이즈(noise)를 가지는 노말 데이터를 이용해서 곡면 복원을 하는 것은 어려운 일이다[18].

기존의 개발된 방법들을 사용하면 제한된 오차 범위 내의 점 집합으로부터 양질의 곡면을 복원할 수 있다. 그러나, 대부분의 기존 방법들은 기하에 기반을 둔 결정론적인 알고리즘(deterministic algorithm)이다. 이러한 방법은 노이즈나 이

상치가 많은 불완전한 데이터가 주어진 경우에 사용자의 작업이 없는 만족할 만한 결과를 얻을 수 없다. 본 논문에서는 주어진 입력 데이터의 부분샘플을 구하는 과정을 이용하여 결정론적인 방법을 확률에 기반을 둔 방법으로 바꾼다. 또, 이러한 방법을 이용하여 이상치나 노이즈가 있어도 안정되게 동작하는 알고리즘을 제안한다.

노이즈의 성질이 확률적이기 때문에 노이즈를 다루기 위해서는 확률적인 방법을 사용하는 것이 효과적인 것이다. 그러나, 확률적인 방법을 사용하면 결정론적인 방법을 사용할 때에 비해 계산량이 많아지게 된다. 이와같은 단점에도 불구하고, 확률적인 방법은 완전히 자동화해서 곡면 복원을 할 수 없는 경우에 사용자의 작업시간을 절약할 수 있다는 장점을 가진다. 또한, 곡면 복원은 한 번만 하면 되는 과정이므로 계산량을 줄이는 일보다는 좋은 결과를 얻는 일이 더 중요하다는 점도 고려되어야 한다.

결정론적인 알고리즘을 확률적인 형태로 바꾸기 위해서는 입력 데이터에서 무작위로 부분샘플링(subsampling)해서 여러 개의 부분집합을 만드는 과정이 필요하다. 이 부분집합은 서로 겹쳐질 수도 있다. 그리고, 이 부분집합 각각을 결정론적인 방법으로 독립적으로 처리하여 몇 개의 앙상블(ensemble)이라고 불리는 서로 다른 결과를 얻는다. 그리고, 이 앙상블들을 종합하여 각각의 독립적으로 처리된 결과보다 더 좋은 결과를 얻는다.

본 논문에서는 결정론적 방법으로 만들어진 앙상블을 이용해서 노말을 구하거나 곡면 복원을 한다. 먼저 노말을 구할 때는 결정론적 방법으로 Hoppe 등 [11]이 제안한 방법을 사용한다. 이 방법은 그래픽스 분야에서 널리 사용되어지는 방법이다 [20] [23]. 곡면 복원을 위해서는 결정론적 방법으

로 Multi-level Partition of Unity (MPU) implicit [18]을 사용한다. 이 방법은 기존 기법 중 최신의 연구이고 계산 속도가 빠른 방법이다.

본 논문의 주요 공헌은 다음과 같다.

- 결정론적 방법에 적용할 수 있는 앙상블이라고 불리는 통계적인 기법의 프레임 워크(framework)를 제시한다. 알고리즘의 안정성을 향상하기 위해 이와 같은 통계적인 기법으로 노이즈가 많은 점 집합을 처리한다.
- 기존의 노말 추정 알고리즘 [11]과 곡면 복원 알고리즘 [18]을 이용하여 실험을 해서 제시한 프레임 워크의 효율성을 보인다.

2. 배경 지식

2.1 앙상블

앙상블 기법은 통계적 학습(statistical learning)의 중요한 주제이다. 일반적으로 앙상블 기법은 같은 데이터에 대해서 여러 번 실행해서 서로 다른 결과를 얻어내고 그것을 하나의 결과로 종합하는 방법이다. 이 방법은 다음 두 가지 과정을 통해서 이루어진다.

- 앙상블의 구성원을 만드는 알고리즘을 최적화
- 앙상블의 구성원을 종합하는 효율적인 방법 개발

이러한 두 과정에 대한 연구는 알고리즘이 감독적(supervised)인지 비감독적(unsupervised)인지에 따라 나뉘어진다. 감독적 알고리즘의 경우에는 결과의 성질에 대한 사전 지식을 가지고 있다. 예를 들자면, 결과의 오차를 아는 경우가 있다. 이러한 사전 지식을 이용해서 오차가 거의 없는 결과를 구할 수 있다 [21, 9].

본 논문에서는 비감독적 알고리즘을 사용한다. 비감독적 알고리즘에서는 결과에 대한 사전 정보를 가지고 있지 않다. 따라서 이론적 측면에서 보면 평균이나 다수 선택(majority voting) 등이 유일한 방법이다. 그럼에도 불구하고 이러한 방법에서 알고리즘을 개선해 좀더 안정된 결과를 얻을 수 있다. 3.2절과 4.2절에서는 이러한 개선점에 대해 설명하고 있다.

[13]에서는 앙상블 기술을 곡면 복원 문제에 적용해서 [12]에서 제안된 알고리즘의 성능을 향상시켰다. [13]에서는 통계적인 알고리즘이 자연스럽게 적용되는 앙상블 기술을 사용하는데 반해, 본 논문에서는 기존의 결정론적인 방법 기반을 둔 통계적인 방법을 사용함으로써 안정성을 향상시킨다.

2.2 노말 추정 기법

점 집합에서의 노말의 추정은 각각의 점에서의 접 평면(tangent plane)을 구하는 과정과 모든 점에서의 접 평면의 일관된 오리엔테이션(consistent orientation)을 구하는 과정으로 나뉘어진다. 이때, 접 평면은 노말의 방향을 나타내고, 접 평면의 오리엔테이션은 노말의 부호를 나타낸다.

접 평면의 추정과 관련해서는 Hoppe 등 [11]이 k 개의 주변 점(k -neighborhood point)에 대한 주 성분 분석(principal component analysis)을 이용했다. 여기서 추정된 접 평면은 k 개

의 주변점의 최소 제곱 근사(least square fitting)이다. Pauly 등 [19]은 최소 제곱 근사를 할때 가중치(weight)를 주어서 이것을 개선했다. Gopi 등 [10]은 k 개의 주변점으로부터의 백터와 추정된 노말과의 내적(inner product)을 최소화하기 위해 특이값 분할(singular value decomposition)을 사용했다. 그밖에 노말 추정과 관련이 있는 연구로는 노말 스무딩(smoothing) [14]을 들 수 있다.

두 번째 과정은 접 평면의 일관된 오리엔테이션을 구하는 과정으로 상대적으로 연구 결과가 많지 않고, 연구 초기의 방법 [11]이 여전히 가장 발전된 기술이다. [11]에서는 임의의 오리엔테이션이 주어진 접 평면에서 시작해서 그것의 오리엔테이션을 주변으로 확산시켜 나간다. 이때 최소 덩어트리(minimum spanning tree)를 사용하게 되고, 접 평면 사이의 각을 트리의 가중치로 사용한다.

접 평면의 오리엔테이션은 두 가지 밖에 없으므로 구하는 과정이 간단해 보인다. 또 대부분의 경우 일관되게 구할 수 있다. 그러나 문제는 오리엔테이션이 일관되지 않게 구해지면 심각한 문제가 발생한다는 것이다. 오리엔테이션이 잘못 구해지면 구해진 노말 백터가 실제 백터와 반대가 되어 오차가 가장 커지게 된다. 실험을 통해 살펴본 결과 두 번째 과정이 최종 곡면 복원의 결과 영상에 더 많은 영향을 준다는 사실을 알 수 있다.

2.3 곡면 복원 기법

곡면 복원 연구의 한 부류는 노말이 없는 점 집합으로부터 곡면을 얻는 연구이다. α -모양(α -shape)을 사용하기도 했고 [3], B-스플라인(B-spline) 패치를 변위 백터(displacement vector)를 근사할 때 사용하기도 했다 [16]. [1, 2, 8, 15]에서는 Delaunay 사면체화(tetrahedrization)를 이용해서 곡면이 만들어 졌다.

곡면 복원 연구의 다른 부류에서는 노말을 가진 점 집합을 입력으로 한다. 이 경우에는 음함수(implicit function) $f: \mathbf{R}^3 \rightarrow \mathbf{R}$ 으로 입력 데이터를 근사하고(fitting), 함수의 0-등위 집합(0-level set)으로부터 곡면을 얻어낸다. 이러한 방법은 [11]에서 시작되었으며, 방사 기저 함수(radial basis function)를 이용하는 방법 [6]이나 MPU implicit 방법 [18]을 포함한다.

본 논문에서는 MPU implicit 방법을 사용한다. 이 방법은 노말을 가진 큰 점 집합으로부터 곡면을 복원하는데에 유용하다. 이 알고리즘은 주어진 공간을 8진트리(octree)를 이용해서 작은 셀(cell)로 나누면서 각각의 공간에서 이차함수로 점들의 부분적인 모양(local shape)을 근사(approximation)한다. 이때, 근사가 충분히 정확하지 않으면 셀을 더 나누게 된다. 이러한 과정을 근사의 오차가 주어진 값보다 작을 때까지 반복한다. 마지막으로 각각의 셀의 근사를 전체 함수가 연속이 되도록 가중치를 주어 더하게 된다.

3. 노말 추정을 위한 앙상블

노말 추정을 위한 앙상블에서는 점 집합 P 를 입력으로 받아 들인다. 먼저, P 의 임의의 부분집합을 만들고, 그것을 사용하여 앙상블의 구성원을 만든다. 이것을 종합하여 각각의 점의 노말을 구하게 된다. 이 과정은 다음과 같이 요약할 수 있다.

노말 추정 앙상블

입력: 점 집합 P .

출력: 노말 P_N 을 가진 점 집합.

1. P 의 중복을 허용하는 부분집합을 구한다.
2. 각각의 부분집합을 처리해서 노말을 구한다.
3. P 의 각각의 점에 대해서, 과정 2에서 구한 노말들을 종합하여 노말을 추정한다.

3.1 앙상블 생성

먼저 무작위로 P 를 부분샘플링해서 부분집합 P_i ($i = 1, \dots, k$)를 만들었다. 결과가 크게 나쁘지 않는 한 무작위 샘플링을 수행하는 여러 가지 방법 중 가장 간단한 방법을 선택한다. 이 실험에서 집합 P_i ($i = 1, \dots, k$)는 같은 수의 점을 갖는다.

$$|P_1| = |P_2| = \dots = |P_k|, \quad (1)$$

단, $|P_i|$ 은 P 에서의 점의 개수이다.

비율 $d = |P_i|/|P|$ 는 부분집합의 평균 밀도이다. d 의 값은 노말 추정 결과에 영향을 준다. $d = 1$ 이면 알고리즘은 결정론적인 알고리즘과 같다. 반대로 d 의 값이 매우 작아지면 부분집합의 밀도가 너무 작아져서 앙상블의 성능이 나빠진다. 실험적으로 $d = 0.2$ 일때 만족할만한 결과를 얻었다.

앙상블의 개수를 k 라 할 때, k 의 선택에 따라 실행시간과 알고리즘의 성능이 달라지게 된다. 크기가 $|P_i|$ 인 점 집합을 이용해 노말 추정 알고리즘을 k 번 돌리게 된다. 여기서 각각의 점에 대해서 $m = k \cdot d$ 개의 노말을 얻을 수 있다. 이때, k 가 커지게 되면 속도가 느려지게 되고, 반대로 k 가 작아지게 되면 얻을 수 있는 노말의 개수가 작아지게 된다. 실험적으로 $k = 35$ 일 때 만족할만한 결과를 얻었으며, 이 경우 각각의 점은 7개의 노말 추정을 갖는다($m = 7$).

일반적으로 무작위적인 샘플링 알고리즘에서는 명시적으로 반복을 피하지 않는다. 따라서, 이러한 방법으로 샘플링을 하면 구해진 각각의 점에 대한 노말의 개수가 일정치 않게 되고, 몇몇 점의 경우는 구해진 노말이 하나도 없을 수도 있다. 이것을 막기 위해 부분집합 P_i 를 구할 때, 반복을 허용하지 않고 샘플링한 후 모든 점이 샘플링된 후에 다시 처음부터 샘플링을 시작한다.

임의의 부분집합 P_i 를 만든 후에, 노말을 추정하기 위해 [11]에 있는 알고리즘을 사용한다. 알고리즘의 코드는 인터넷에서 구할 수 있고 파라미터는 기본 파라미터를 사용한다.

3.2 노말의 평균

주어진 점에서 얻어진 서로 다른 노말의 평균을 구해서, 좀 더 신뢰성이 높은 하나의 노말을 구한다. 첫 번째 방법으로 단순히 벡터의 평균을 구하는 방법이 있다.

$$\mathbf{n} = \sum_{j=1}^m \mathbf{n}_j / \left| \sum_{j=1}^m \mathbf{n}_j \right|, \quad (2)$$

단, \mathbf{n}_j 은 주어진 점에서 서로 다른 노말 추정이다.

식 (2)를 사용하면 데이터의 이상치가 평균에 많은 영향을 주는데도 불구하고 상당히 좋은 결과를 얻을 수 있었다.

노말의 평균을 구할 때, [5]에서 제안된 방법을 사용할 수도 있다. 이 방법에서는 먼저 수식 (2)을 사용해서 근사적인 평균값을 구한다. 근사적으로 구해진 평균값을 중심으로 구의 점을 평균값을 지나는 평면에 대응시키고, 대응된 평면에서 다시 평균값을 구하게 된다. 그리고, 이 평균값을 구의 점으로 대응시키게 되고 다음부터는 대응된 구의 점을 평균값으로 해서 위와 같은 과정을 반복한다. 이러한 알고리즘은 수렴하는 성질이 있다고 알려져 있으며, 수렴하게 되면 알고리즘을 종료한다. 이러한 알고리즘은 이론적으로 좋은 성질을 가지고 있지만, 본 논문의 실험에서는 큰 이점을 가지지는 못했다.

본 논문에서는 기본적으로 [5]에서 제안된 평균을 구하는 방법을 사용하되 이상치를 제외하고 평균을 구하는 방법을 채택했다. 먼저 식 (2)를 사용해서 노말을 추정하고, 추정된 노말 \mathbf{n} 과 각각의 노말 \mathbf{n}_j 와의 각을 θ_j 를 구한다. 이때 일반성을 잃지 않고 다음을 가정할 수 있다.

$$\theta_1 \leq \theta_2 \leq \dots \leq \theta_m \quad (3)$$

각각의 \mathbf{n}_j 에 대해서 θ_j 의 값이 큰 r 개를 제외하고, 평균을 낸다.

$$\mathbf{n}_f = \text{Aver}(\mathbf{n}_1, \dots, \mathbf{n}_{m-r}), \quad (4)$$

(단 Aver())은 [5]를 이용해서 구하는 평균)

이상치 노이즈가 전체 샘플의 절반을 넘을 가능성은 희박하기 때문에, 본 논문의 실험에서는 $r = \lfloor m/2 \rfloor$ 를 사용한다.

4. 곡면 복원을 위한 앙상블

곡면 복원 앙상블의 입력은 노말을 가지고 있는 점 집합이고, 결과는 곡면 S 이다. 알고리즘은 다음과 같이 요약된다.

곡면 복원 앙상블

입력: 노말을 가진 점 집합 P_N .

출력: 곡면 S .

1. P_N 에서 중복을 허용하는 임의의 부분집합을 만든다.
2. 각각의 부분집합을 이용해 곡면을 독립적으로 복원한다.
3. 과정 2에서 얻어진 곡면을 종합하여 하나의 곡면 S 를 만든다.

4.1 앙상블 생성

기본적으로 4.1절에서 사용한 샘플링 방법을 사용하지만, 한 가지 차이점이 있다. 4.1절에서는 각각의 점이 가지는 노말 앙상블의 구성원의 개수가 같아야 하므로, 각각의 점을 같은 횟수만큼 샘플링해야 한다. 그러나, 곡면의 복원을 위한 앙상블에서는 각각의 점을 같은 횟수만큼 샘플링할 필요가 없다. 이러한 이유에서 본 절에서는 단순히 P_N 으로부터 임의로 부분샘플링을 해서 부분집합을 구하는 과정을 반복한다.

임의의 부분집합을 구한 후에 각각의 부분집합의 곡면을 생성하기 위해 MPU 알고리즘을 이용한다. MPU 알고리즘은 인터넷에서 구할 수 있고, 다각형화(polygonization)를 위한 것을 제외하고는 기본 파라미터를 사용한다.

4.2 곡면 평균

4.1절에서 m 개의 함수를 얻을 수 있다.

$$f_j: \mathbf{R}^3 \rightarrow \mathbf{R}, \quad j = 1, \dots, m. \quad (5)$$

각각의 함수 f_i 의 0-등위 집합이 곡면 S_j 를 정의한다. 앙상블 곡면 S 를 얻기 위해서는 각각의 함수 f_i 의 평균을 내서 하나의 함수 f 를 구해야 한다.

3.2절과 비슷한 방법으로 각각의 함수 f_i 의 단순한 평균을 구하는 방법이 있다.

$$f(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^m f_j(\mathbf{x}). \quad (6)$$

그러나, 이와 같은 방법은 이상치에 의한 영향을 많이 받기 때문에 좀더 안정된 결과를 얻기 위해서 이상치를 제외하고 값들의 평균을 구한다. 일반성을 잃지 않고, 점 \mathbf{x} 에 대해서 다음을 가정할 수 있다.

$$f_1(\mathbf{x}) \leq f_2(\mathbf{x}) \leq \dots \leq f_m(\mathbf{x}). \quad (7)$$

이때, \mathbf{x} 에서의 평균을 다음과 같이 계산한다.

$$f(\mathbf{x}) = \frac{1}{m-2r} \sum_{j=r+1}^{m-r} f_j(\mathbf{x}). \quad (8)$$

다시 말하면, \mathbf{x} 에서 가장 작은 r 개와 가장 큰 r 개를 제외하고, 나머지 f_i 를 평균을 낸다. 여기에서도 노말 앙상블과 비슷하게 함수 값들의 반을 제거하기 위해서 $r = \lfloor m/4 \rfloor$ 을 사용한다.

수식 (8)에서 f_i 가 연속이므로 f 의 불연속은 f_i 의 크기의 순서가 바뀔 때만 나타날 수 있다. 그러나 f_i 의 순서가 바뀔 때 바뀌는 두 함수의 함수값이 같으므로, f 는 연속함수이다. 6절에서 볼 수 있는 결과를 얻기 위해서는 격자점에서 함수 f 의 평균을 구하고, 마칭 큐브(marching cube) 알고리즘 [17]을 사용해서 f 의 0-등위 집합을 근사한다.

5. 검증 실험

앙상블 알고리즘을 검증하기 위해 다음과 같은 함수가 나타내는 곡면으로 실험을 하였다.

$$x^4 - 5x^2 + y^4 - 5y^2 + z^4 - 5z^2 + 11.8 = 0, \quad (9)$$

이 곡면은 그림 1에서 보듯이 검증하기에 충분히 복잡한 형태를 가지고 있다. 이 곡면은 tangle cube라고 불린다.

검증 실험에서는 [15]에서 사용된 방법을 이용했다. tangle cube로부터 $N = 244,936$ 개의 점을 샘플링한 후에 N 의 30%에 해당하는 점에 노이즈를 주어 추가했다. 여기서 노이즈의 최대값은 경계 상자(bounding box)의 대각선의 길이의 7%로 하였다.

5.1 노말 앙상블 실험

노말 추정에서는 [11]의 방법과 수식 (2)의 단순 평균 방법과 수식 (4)의 안정된 평균 방법을 비교했다. 여기서는 노이즈가 주어진 점을 제외한 모든 점에서 공식에 의해 구해진 정확한 노말과 알고리즘에 의해 구해진 노말 사이의 오차를 구했다.

표 1의 단일 노말 추정 결과와 노말 추정 결과에서 다음 식으로 구한 평균 제곱 오차를 보여준다.

$$RMS = \sqrt{\frac{1}{N} \sum_j (1 - \mathbf{n}_j^e \cdot \mathbf{n}_j)^2} \quad (10)$$

그리고, 최대 오차는 다음 식으로 구한다.

$$MAX = \max(1 - \mathbf{n}_j^e \cdot \mathbf{n}_j)^2 \quad (11)$$

단, \mathbf{n}_j^e 은 추정된 노말이고, \mathbf{n}_j 은 정확한 노말이다.

앙상블의 속도는 단일 노말 추정에 비해 10배 정도 느리다. 이러한 단점에도 불구하고 오차가 10배 정도 감소하는 효과를 얻을 수 있다. 그림 1의 가장 위 행은 실험 점 집합의 렌더링을 보여준다.

추정된 노말을 테스트하기 위해서 주어진 노말을 단일 MPU 알고리즘을 이용해서 곡면 복원을 했다. 그리고, 메트로(Metro tool) [7]를 이용해서 복원된 곡면과 원래 곡면 사이의 거리를 측정했다. 표 1에 단일 MPU에서 볼 수 있듯이 앙상블을 사용하면 오차는 거의 절반이 되었다. 그리고, 수식 (4)을 이용한 평균은 단순한 평균에 비해 조금 나은 결과를 보였다.

그림 1의 마지막 행은 세 가지 단일 MPU 복원의 렌더링을 보여준다. 정밀한 노말 정보가 곡면 복원의 질에 많은 영향을 줄 수 있다.

5.2 곡면 앙상블 실험

그림 2에서는 구해진 노말을 이용해서 복원된 곡면을 볼 수 있다. 안정된 노말 평균 방법을 사용하면 더 정확한 노말을 구할 수 있기 때문에, 곡면 앙상블을 위한 실험에서는 안정된 노말 평균 방법을 이용해서 구한 노말을 실험에 사용했다. 먼저 그림 2(b)는 단일 MPU를 이용한 결과이다. 이것은 곡면 앙상블을 하지 않았을 때, 노말 앙상블의 한계를 보여준다. 그림 2(c)와 2(d)는 각각 수식 (6)과 (8)을 이용해서 곡면 앙상블을 한 결과를 보여준다. 이것은 노말 앙상블과 곡면 앙상블을 함께 사용하면 더 좋은 결과를 얻을 수 있음을 보여준다.

메트로를 이용해서 그림 2(c)와 2(d)에서 보여진 복원된 결과와 원래 곡면 사이의 거리를 잴다. 그 결과는 표 1에 곡면 앙상블 열에서 볼 수 있다. 여기서 수식 (8)의 평균이 수식 (6)의 평균보다 더 좋다. 그러나, 영상에서 보이는 것과 같은 큰 차이는 없음을 알 수 있다. 그 이유는 단일 MPU가 더 촘촘한 데이터를 이용해서 곡면을 만들기 때문에 그림 2(b)에서 나타난 잘못된 부분 이외의 부분에서는 앙상블보다 근사가 더 잘 되기 때문이다.

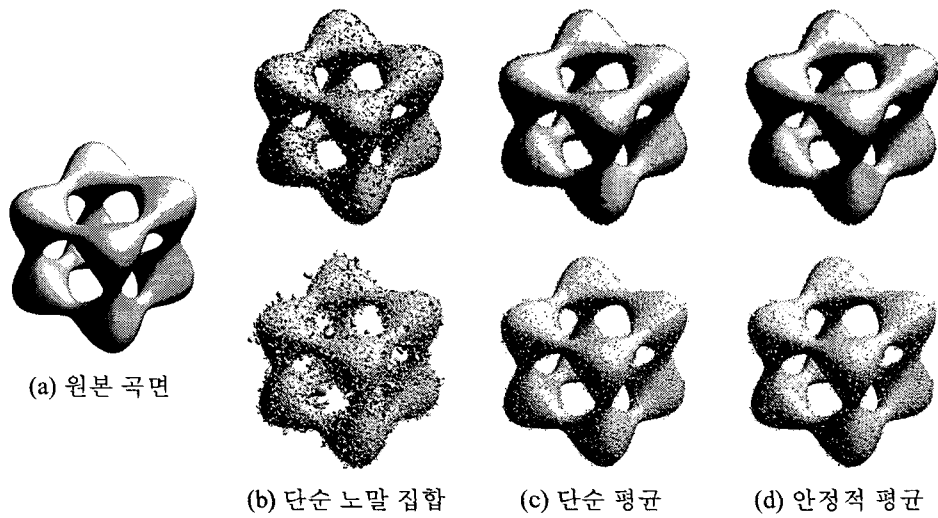


그림 1: Tangle Cube: 서로 다른 노말 추정과 노말로부터 얻은 곡면의 비교. (b), (c), (d)의 첫 번째 행에서 각각의 점은 노말 벡터를 사용해서 밝기를 결정해 렌더링한다. (b), (c), (d)의 두 번째 행은 단일 MPU를 사용해서 복원된다.

모델	점의 수	구분	단일 노말 추정	노말 앙상블		단일 MPU			곡면 앙상블 (그림 2)	
				단순 평균	안정적 평균	단일 노말 추정	단순 평균	안정적 평균	단순 평균	안정적 평균
Tangle cube	317,934	시간 (sec)	69.2	658.61	660.69	50	44	45	171	171
		RMS ($\times 10^{-3}$)	18.0	2.90	2.78	7.89	3.87	3.87	4.08	2.60
		최대 ($\times 10^{-3}$)	1289.0	54.2	50.4	43.48	31.46	29.94	30.17	26.26

표 1: 그림 1(a)에서 참고하는 곡면으로부터의 계산 시간과 오차. 노말의 계산 시간은 2.0 GHz Pentium 4 프로세서와 1 GB 메모리를 가진 리눅스 PC에서 초단위로 계산된 시간이고, 곡면 복원은 3.2 GHz Pentium 4 프로세서와 2 GB 메모리를 가진 윈도우 PC에서 계산된 시간이다.

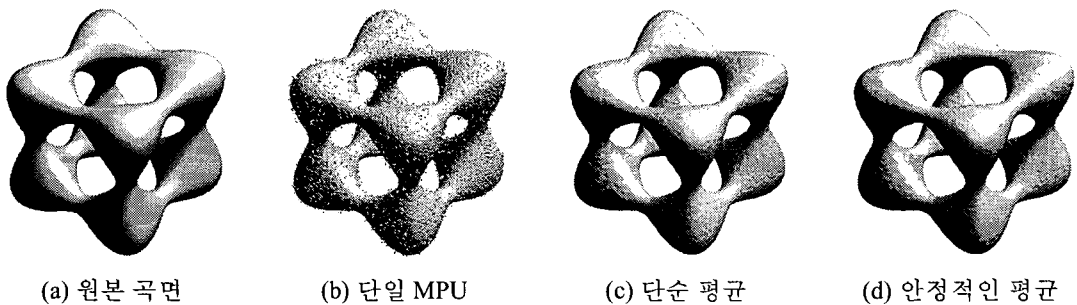


그림 2: Tangle Cube: 단일 MPU와 앙상블을 사용해 얻은 곡면의 비교

6. 실험 결과

이 절에서는 잘 알려진 점 집합으로부터 실험결과를 얻었다. 여기에 사용된 모델은 ViHAP3D 프로젝트에서 얻은 bierkrug와 karaffe 모델, Buddha 모델, Stanford 3D 스캐닝 저장소로부터 얻은 bunny 모델와 armadillo 모델이다. 이 절에서는 노말 이상불과 곡면 이상불을 위해 각각 수식 (4)와 (8)을 사용했다.

그림 3에서는 이상불을 했을 때와 단일 노말 추정을 사용했을 때 노말 추정 결과를 보여주고, 이것에 단일 MPU 알고리즘을 수행한 결과를 비교했다. 그림 3(a)와 3(b)를 보면 bierkrug를 이상불 했을 때보다, 단일 노말 추정을 했을 때, 조각된 부분에서 세부형태가 더 살아있는 것처럼 보인다. 하지만, 이것이 단일 노말 추정을 한 결과가 이상불 결과보다 좋다는 것을 뜻하지는 않는다. 이것은 그림 3(c)와 3(d)를 보면 알 수 있다. 어느 쪽 결과이던지 곡면 복원 후에는 조각된 부분이 스무딩 됨을 알 수 있다.

그림 4(a), 4(c), 4(e)와 4(f)는 그림 3에서 단일 MPU 알고리즘을 적용해서 곡면 복원한 결과를 확대해서 보여주고 있다. 그림 4(b), 4(d), 4(g)와 4(h)는 그림 3에서 각각의 노말에 대해서 곡면 복원 이상불 알고리즘을 적용한 결과를 확대해서 보여주고 있다. 노말과 곡면 이상불을 동시에 한 것이 가장 좋은 결과를 보여주고 있다. 특히, 그림 4(d)나 그림 4(h)에서 볼 수 있듯이 이 방법을 사용하면 bierkrug의 밑 부분의 잘못된 부분을 제거하거나 손잡이 부분을 제대로 복원할 수 있다.

그림 5에서는 karaffe 모델을 위한 단일 알고리즘과 이상불 알고리즘의 서로 다른 조합을 비교한다. 여기서는 한 가지 예외적인 현상이 발생한다. 단일 노말 추정을 사용한 곡면 이상불이 이상불 노말을 사용한 곡면 이상불보다 더 좋은 결과를 나타낸다. 그 이유는 karaffe의 윗 부분의 점들이 이미듬성듬성하게 존재하기 때문에 이것의 부분집합은 너무 적은 정보를 가지기 때문이다.

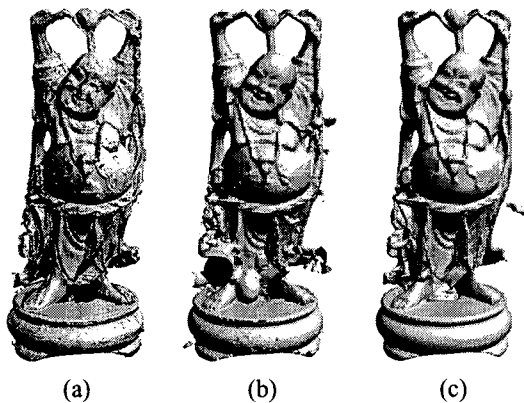


그림 6: Buddha: (a) 입력 점 집합. (b) 단일 MPU 복원. (c) MPU 이상불 복원.

그림 6에서는 입력 데이터와 Buddha모델의 복원 결과를 보여준다. 여기서는 노말의 경우는 단순 노말 추정 방법을 사용했고, 곡면 이상불의 경우는 MPU 이상불을 사용했다. 그림 7에서 Buddha의 11개의 MPU 이상불을 볼 수 있다.

그림 8은 armadillo 모델의 점 집합과 단일 MPU 복원과 이상불 복원 결과를 보여준다. 노말은 단일 및 이상불 노말 추

정을 통해서 각각 얻어진다. 그림 8(f)는 노말 이상불과 곡면 이상불을 사용하여 얻은 결과로써, 가장 좋은 결과를 보여주고 있다. 여기서 이상불 방법을 사용했을 때 중요한 기하학적인 세부형태가 보존됨을 알 수 있다. 여기서 사용한 입력 데이터는 처리되지 않은 원래 데이터이고, 따라서, 미리 처리되어진 깨끗한 데이터를 사용한 곡면 복원과 비교할 필요는 없다.

표 2는 모델의 크기와 이상불에 관한 정보와 수행시간을 보여준다.

7. 결론 및 향후 연구

본 논문에서는 이상불 기법을 크기가 크고, 노이즈가 많은 데이터로부터 곡면을 복원하는 문제에 적용했다. 이상불 기법을 곡면 복원을 위해 노말을 추정하는 문제나 노말을 갖는 점 집합으로부터 곡면 복원을 하는 문제에 효과적으로 적용했다.

본 논문에서는 노말 추정과 곡면 복원 문제를 독립적으로 연구했다. 그러나, 노말 추정한 결과를 곡면 복원 이상불에 적용시키는 실험도 했다. 이 두 가지 과정을 모두 사용함으로써 더욱 좋은 곡면 복원 결과를 얻을 수 있었다. 이것은 첫 번째 과정이 노말 이상치를 제거하고, 두 번째 과정이 점의 위치의 이상치를 제거하기 때문에 나타나는 현상이다.

이상불 알고리즘이 노이즈에 덜 민감하게 만들어졌기 때문에 스무딩과 비슷한 점이 많다. 여기서 대부분의 스무딩 알고리즘에서 사용하는 주변점들의 범위는 이상불에서의 점들의 부분집합의 밀도와 관련이 있다. 그러나, 스무딩에 비해 이상불이 가지는 장점이 있다. 먼저, 노이즈가 확률적으로 존재하기 때문에 확률적인 방법이 노이즈를 더 잘 해결할 수 있을 것이라고 기대할 수 있다. 그리고, 스무딩은 세부형태를 잃게 되지만, 이상불 알고리즘에서는 이상불의 구성원의 수를 증가시킴으로써 세부형태를 보존할 수 있다.

스무딩에 비해 이상불이 가지는 가장 큰 장점은 이상치의 영향을 적게 받는다는 것이다. 평균을 내기 전에 이상치를 제거함으로써, 노말이 잘못된 방향으로 구해지는 것을 막고 점의 잘못된 위치가 최종 곡면에 영향을 주는 것을 막는다. 반대로 이상치를 제거하지 않고 복원된 곡면은 왜곡이 매우 심해 스무딩으로 해결되지 않는다.

향후 연구 과제로는 이러한 확률적인 기법을 다른 곡면 복원 알고리즘에도 적용하는 것이다. 또한, 이상불의 수와 크기가 결과에 어떤 영향을 주는지에 대한 이론적인 분석도 중요한 향후 연구 과제이다.

감사의 글

MPU의 성질에 대한 의견을 주고, MPU 이상불의 구현에 많은 도움을 준 Yutake Ohtake에게 감사드립니다. Buddha와 armadillo 모델은 Stanford 그래픽스 연구실에서 구한 것입니다. 본 연구는 BK21사업을 통하여 포항공과대학교 전자, 컴퓨터공학부에 주어진 교육부의 재정 지원, 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터 육성 및 지원사업의 지원을 얻어 수행되었습니다.

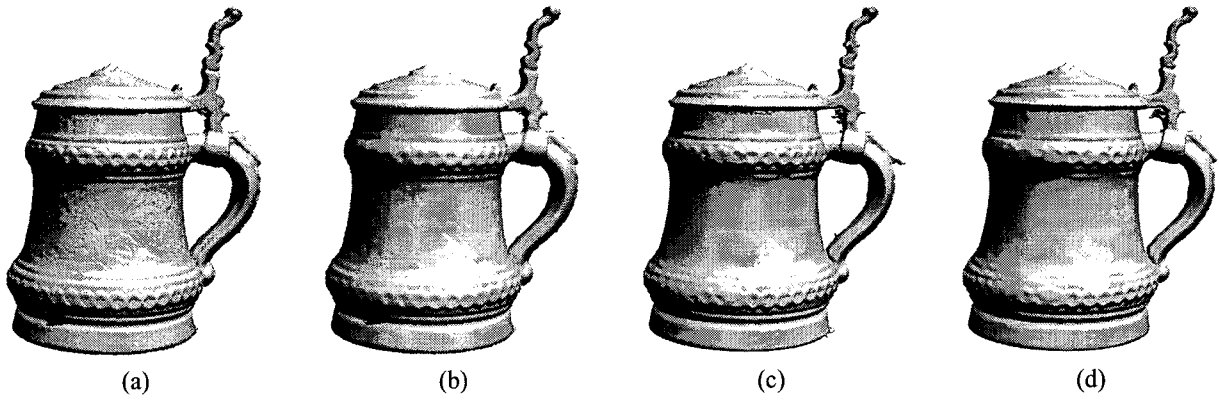


그림 3: Bierkrug: (a)단일 추정에 의해 얻어진 노말. (b) 앙상블에 의해 얻어진 노말. (c)와 (d): 각각 (a)와 (b)의 단일 MPU 복원 결과.

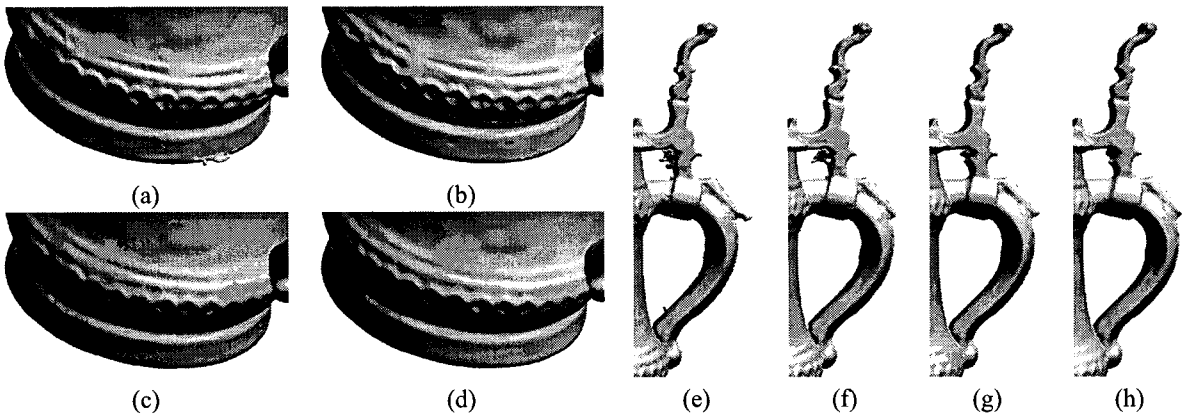


그림 4: Bierkrug: (a)와 (e): 그림 3(c)의 확대. (c)와 (f): 그림 3(d)의 확대. (b)와 (g): 그림 3(a)에 곡면 앙상블을 적용한 결과. (d)와 (h): 그림 3(b)에 곡면 앙상블을 적용한 결과.

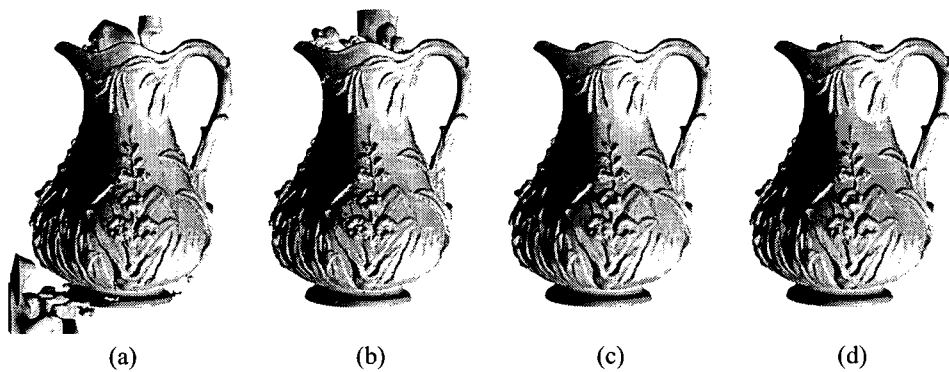


그림 5: Karaffe: (a) 단일 노말 추정과 단일 MPU 복원의 조합. (b) 앙상블 노말과 단일 MPU 복원의 조합 (c)와 (d): 각각 (a)와 (b)의 노말 집합을 사용한 곡면 앙상블.

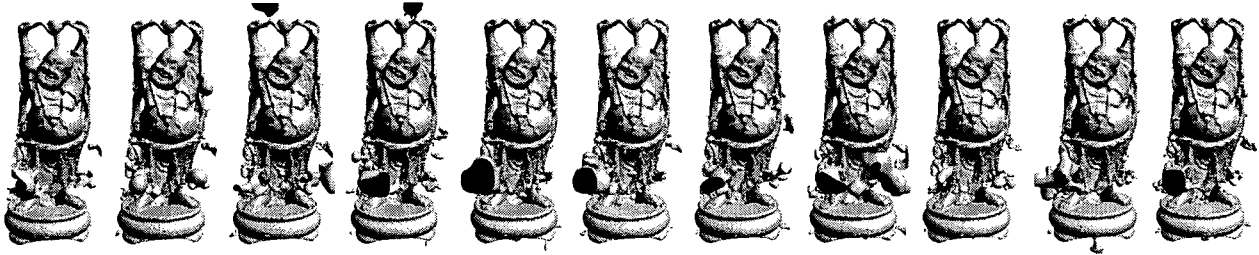


그림 7: Buddha: 그림 6에서 Buddha의 11개의 곡면 앙상블 구성원.

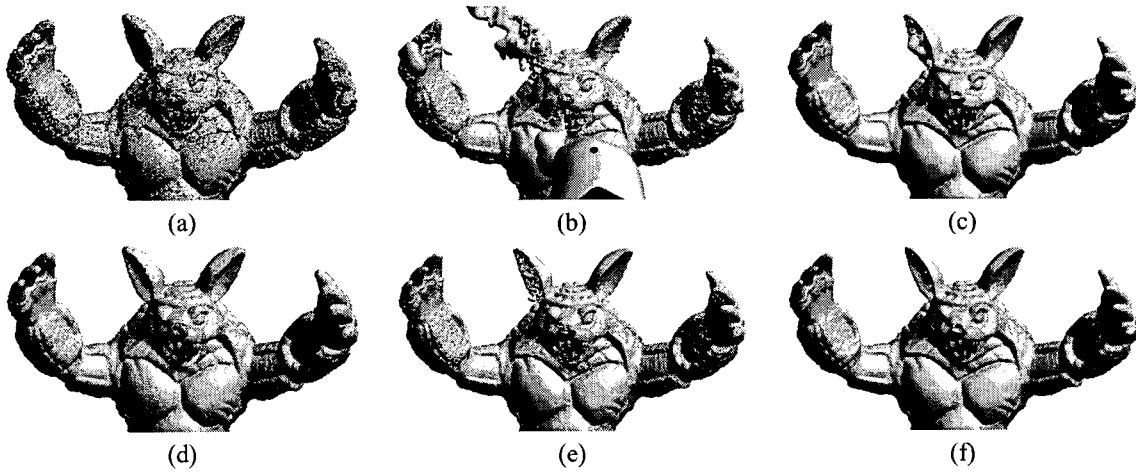


그림 8: Armadillo: (a)-(c) 단일 추정 노말을 사용한 결과. (d)-(f) 앙상블 추정 노말을 사용한 결과. 왼쪽: 노말을 포함한 점. 중앙: 단일 MPU. 오른쪽: MPU 앙상블

모델	점의 수 (합계)	노말 추정			곡면 복원		
		단순 노말	점의 수 (구성원)	노말 앙상블	단순 MPU	점의 수 (구성원)	곡면 앙상블
Bierkrug	500,690	129.84	100K	1251.5	44	50K	162
Karaffe	1,001,915	341.61	200K	3287.45	75	100K	261
Buddha	3,260,401				146	330K	250
Armadillo	1,394,271	651.03	280K	6326.55	117	140K	258

표 2: 계산 시간 통계: 노말 추정과 곡면 복원을 위한 시간은 초 단위로 측정 되었다. 컴퓨터의 사양은 표 1과 같다.

참고 문헌

- [1] AMENTA, N., BERN, M., AND KAMVYSSELIS, M. A new voronoi-based surface reconstruction algorithm. In *SIGGRAPH 98, Conference Proceedings* (1998), pp. 415–422.
- [2] AMENTA, N., CHOI, S., AND KOLLURI, R. K. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications* 19, 2-3 (2001), 127–153.
- [3] BAJAJ, C. L., BERNARDINI, F., AND XU, G. Automatic reconstruction of surfaces and scalar fields from 3D scans. In *SIGGRAPH 95, Conference Proceedings* (1995), pp. 109–118.
- [4] BERNARDINI, F., AND RUSHMEIER, H. The 3D model acquisition pipeline. *Computer Graphics Forum* 21(2) (2002), 149–172.
- [5] BUSS, S. R., AND FILLMORE, J. P. Spherical averages and applications to spherical splines and interpolation. In *ACM Transactions on Graphics*, vol. 20(2). 2001, pp. 95–126.
- [6] CARR, J. C., BEATSON, R. K., CHERRIE, J. B., MITCHELL, T. J., FRIGHT, W. R., MCCALLUM, B. C., AND EVANS, T. R. Reconstruction and representation of 3d objects with radial basis functions. In *SIGGRAPH 01, Conference Proceedings* (2001), pp. 67–76.
- [7] CIGNONI, P., ROCCHINI, C., AND SCOPIGNO, R. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum* 17, 2 (1998), 167–174.
- [8] DEY, T. K., AND GOSWAMI, S. Tight cocone: A watertight surface reconstructor. *J. Comput. Inf. Sci. Eng.* 3, 4 (2003), 302–307.
- [9] FREUND, Y., AND SCHAPIRE, R. E. Experiments with a new boosting algorithm. In *International Conference on Machine Learning* (1996), pp. 148–156.
- [10] GOPI, M., KRISHNAN, S., AND SILVA, C. T. Surface reconstruction based on lower dimensional localized delaunay triangulation. In *Computer Graphics Forum (Eurographics 2000)* (2000).
- [11] HOPPE, H., DEROSE, T., DUCHAMP, T., McDONALD, J., AND STUETZLE, W. Surface reconstruction from unorganized points. In *SIGGRAPH 92, Conference Proceedings* (1992), pp. 71–78.
- [12] IVRISIMTZIS, I., JEONG, W.-K., LEE, S., LEE, Y., AND SEIDEL, H.-P. Neural meshes: Surface reconstruction with a learning algorithm. Tech. Rep. MPI-I-2004-4-005, Max-Planck-Institut für Informatik, Saarbrücken, 2004.
- [13] IVRISIMTZIS, I., LEE, Y., LEE, S., JEONG, W.-K., AND SEIDEL, H.-P. Neural mesh ensembles. In *3D Data Processing, Visualization, and Transmission, 2nd International Symposium on (3DPVT'04)* (Thessaloniki, Greece, 2004), Y. Aloimonos and G. Taubin, Eds., IEEE.
- [14] JONES, T. R., DURAND, F., AND ZWICKER, M. Normal improvement for point rendering. *IEEE Computer Graphics and Applications* 24, 4 (2004), 53–56.
- [15] KOLLURI, R., SHEWCHUK, J. R., AND O'BRIEN, J. F. Spectral surface reconstruction from noisy point clouds. In *Proceedings of the Symposium on Geometry Processing (SGP-04)* (2004), D. Fellner and S. Spencer, Eds., pp. 11–22.
- [16] KRISHNAMURTHY, V., AND LEVOY, M. Fitting smooth surfaces to dense polygon meshes. In *SIGGRAPH 96, Conference Proceedings* (1996), pp. 313–324.
- [17] LORENSEN, W. E., AND CLINE, H. E. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of SIGGRAPH 87* (1987), pp. 163–168.
- [18] OHTAKE, Y., BELYAEV, A., ALEXA, M., TURK, G., AND SEIDEL, H.-P. Multi-level partition of unity implicit. In *SIGGRAPH 2003, Conference Proceedings* (2003), pp. 463–470.
- [19] PAULY, M., KEISER, R., KOBELT, L. P., AND GROSS, M. Shape modeling with point-sampled geometry. In *Proceedings of SIGGRAPH 03* (2003), pp. 641–650.
- [20] SAINZ, M., PAJAROLA, R., MERCADE, A., AND SUSIN, A. A simple approach for point-based object capturing and rendering. *IEEE Computer Graphics and Applications* 24, 4 (2004), 24–33.
- [21] SCHAPIRE, R. E. The strength of weak learnability. *Machine Learning* 5 (1990), 197–227.
- [22] WEYRICH, T., PAULY, M., HEINZLE, S., KEISER, R., SCANDELLA, S., AND GROSS, M. Post-processing of scanned 3d surface data. In *Symposium on Point-Based Graphics* (2004), pp. 85–94.
- [23] XU, H., AND CHEN, B. Stylized rendering of 3d scanned real world environments. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering* (2004), ACM, pp. 25–34.