

## Efficient GPU Framework for Adaptive and Continuous Signed Distance Field Construction, and Its Applications

Jong-Hyun Kim\*

\*Professor, School of Software Application, Kangnam University, Yongin, Korea

### [Abstract]

In this paper, we propose a new GPU-based framework for quickly calculating adaptive and continuous SDF(Signed distance fields), and examine cases related to rendering/collision processing using them. The quadtree constructed from the triangle mesh is transferred to the GPU memory, and the Euclidean distance to the triangle is processed in parallel for each thread by using it to find the shortest continuous distance without discontinuity in the adaptive grid space. In this process, it is shown through experiments that the cut-off view of the adaptive distance field, the distance value inquiry at a specific location, real-time raytracing, and collision handling can be performed quickly and efficiently. Using the proposed method, the adaptive sign distance field can be calculated quickly in about 1 second even on a high polygon mesh, so it is a method that can be fully utilized not only for rigid bodies but also for deformable bodies. It shows the stability of the algorithm through various experimental results whether it can accurately sample and represent distance values in various models.

▶ **Key words:** Compute unified device architecture, Adaptive distance field, Data visualization, Collision handling, Rendering

### [요 약]

본 논문에서는 적응형 및 연속적인 부호 거리장을 빠르게 계산하기 위한 새로운 GPU 기반 프레임워크를 제안하고, 이를 활용한 렌더링/충돌처리 관련 사례를 살펴본다. 삼각형 메쉬로부터 구성된 쿼드트리를 GPU 메모리로 전달하고, 이를 활용하여 삼각형에 대한 유클리디안 거리를 각 스레드 별로 병렬 처리함으로써 적응형 격자 공간에서 불연속 없이 연속적인 최단 거리를 찾는다. 이 과정에서 적응형 부호 거리장의 절단면 보기, 특정 위치에서의 거리 값 조회, 실시간 레이트레이싱 및 충돌처리 작업을 빠르고 효율적으로 수행될 수 있는지를 실험을 통해 보여준다. 제안하는 방법을 사용하면 하이폴리곤 메쉬에서도 1초 내외로 빠르게 적응형 부호 거리장을 계산할 수 있기 때문에 강제뿐만 아니라 변형체에서도 충분히 활용될 수 있는 방법이며, 다양한 모델에서도 정확하게 샘플링하고 거리 값을 나타낼 수 있는지 다양한 실험 결과를 통해 알고리즘의 안정성을 보여준다.

▶ **주제어:** 쿼드, 적응형 부호 거리장, 데이터 시각화, 충돌처리, 렌더링

- First Author: Jong-Hyun Kim, Corresponding Author: Jong-Hyun Kim
- \*Jong-Hyun Kim (jonghyunkim@kangnam.ac.kr), School of Software Application, Kangnam University
- Received: 2022. 02. 15, Revised: 2022. 03. 08, Accepted: 2022. 03. 21.
- This paper is an extension of the paper presented("Optimized construction and visualization of GPU-based adaptive and continuous signed distance field, and its applications") at the Conference(Proceedings of the Korea Society of Computer and Information Conference 2021, Best Paper Award).

## I. Introduction

거리장은 컴퓨터그래픽스에서 널리 사용되고 있다 : 물리 기반 시뮬레이션[1,2], 지오메트리 프로세싱[3,4], 충돌 처리[5,6], 데이터 시각화[7,8] 등. 거리장을 구성하는데 있어서 주요 병목 현상은 쿼리 지점에서 3차원 모델까지의 최단 거리를 결정하는 과정이다. 최적화 자료구조 중 하나인 kd-tree는 일반적으로 이 과정을 가속화하는 데 사용되지만[9], 쿼리 지점이 많은 경우 이 접근 방식은 계산 비용이 많이 든다. 이 문제에 대한 한 가지 접근 방식은 미리 샘플링된 거리값을 저장하고 그 값을 보간하는 방식이다 [10,11] (Fig. 1 참조).

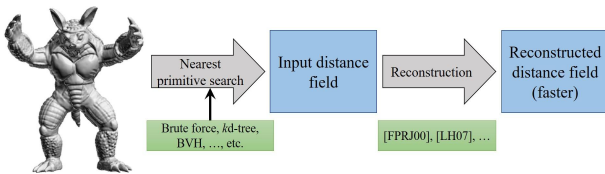


Fig. 1. Overview of distance-field reconstruction algorithms.

Nyquist-Shannon 샘플링 이론에 따르면 공간 샘플링 간격 크기의 두배 보다 작은 특징이 손실된다고 주장한다 [2]. 하지만, 매우 정밀한 샘플링은 저장 및 계산 측면에서 비용이 많이 소요된다. 일반적으로 선형 보간에 의해 발생하는 오류는 최종 렌더링 과정 후에 이질적인 문제를 초래한다. 이러한 문제는 샘플링 해상도를 높여도 완전히 해결되긴 어려우며, 공간 해상도를 높이면 자연스럽게 완화될 수는 있지만 계산량이 커지게 된다.

샘플링된 값의 보간을 이용하는 기존 방법은 부호 거리장에서만 적용될 수 있는 접근법이지만, 3D 모델에 따라서 구멍이나 T-junctions 문제가 나타날 수 있다. 이 문제는 연결정보가 없는 Polygon soup 모델에서 더욱더 심각하게 나타난다.

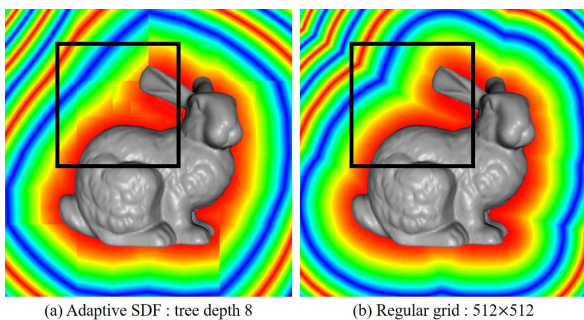


Fig. 2. Comparison of distance field quality of adaptive tree and regular grid : discontinuity problem[10].

본 논문에서는 각 노드에 가장 가까운 거리 값을 저장하고, Primal tree[11]와 같은 자료구조를 GPU 프레임워크에서 빠르게 계산할 수 있는 방법을 제시한다. 제안하는 방법은 이해하고 구현하고 쉽지만, 이전 방법보다 빠른 성능을 보여준다. 또한, 논 매니폴드 메쉬(Non-manifold mesh)에서 부호없는 거리(Undsigned distance)를 나타낼 수 있으며, 적응형 트리에서 일반적으로 나타나는 불연속적인 문제를 완화시킨다. Fig. 2b에서 보듯이 정규격자에서는 연속적인 필드가 나타나지만, 적응형 트리에서는 노드간의 차이로 인해 불연속적인 문제가 나타나며(Fig. 2a 참조), 본 논문에서는 이러한 문제를 완화시킬 수 있는 결과를 보여준다.

본 논문에서는 CUDA(Compute Unified Device Architecture)를 기반으로 적응형 부호 거리장을 생성하는 간단하지만 강력한 방법을 제안한다. 우리의 접근 방식은 최근 GPU 아키텍처에서 사용할 수 있는 엄청난 수의 스레드를 효율적으로 활용하여 이전 GPU 기반 부호 거리장 기법보다 빠른 결과를 실험을 통해 보여준다. 각 스레드에서 가장 가까운 거리 값을 찾기 위해 공간분할 데이터 구조는  $k$ -d tree를 사용하여 GPU의 성능을 효율적으로 개선시킨다. 지오메트리 프로세싱에서 사용하는 공간 분할 계층 알고리즘을 사용하면 샘플 위치로부터 가까운 삼각형 집합을 빠르게 얻어낼 수 있기 때문에 가속화가 가능하다. 그러나 이들의 효율성은 삼각형 정보가 아닌 OBB(Oriented bounding box) 또는 AABB(Axis-aligned bounding box)를 사용하기 때문에 정확성 측면에서 종종 불만족스러운 결과를 만들어내기도 한다. Park 등은 CUDA를 활용하여 부호 거리장을 빠르게 구축할 수 있는 방법을 제시했다[13]. 이들은 복잡한 알고리즘 디자인을 피하기 위해 GPU에서 제공하는 많은 개수의 스레드를 단순하게 활용하여 부호 거리장의 속도 문제를 해결하였다. 하지만, 하이폴리곤 모델인 경우, 삼각형 개수가 100만개 이상일 경우 이 알고리즘은 효율성이 감소될 수밖에 없는 구조이다. 본 논문에서는 이러한 비효율성을 피하고자 새로운 GPU 프레임워크를 제안한다:

- 1) 하이폴리곤 메쉬에서 공간 쿼리를 가속화 할 수 있는 GPU 기반  $k$ -d tree 순회 기법
- 2) GPU 기반으로 최적화된 Primal tree[11] 구축 기법

## II. Preliminaries

### 1. Related work

거리장을 샘플링하는 브루트 포스(Brute-force)접근 방식은  $n$ 개의 삼각형들에 대해서 모든 거리 값을 측정하고 최소값을 찾는다. 이 방법은  $O(n)$ 의 복잡성을 가지므로  $n$ 이 클 때 합리적인 계산 시간을 보장할 수 없다.  $k$ -d tree는 효과적으로 가장 가까운 이웃 검색 문제를 풀 수 있는 자료구조이기 때문에 일반적으로 많이 활용되고 있는 방식이며, 이것을 활용하면 알고리즘 복잡도가  $O(\log n)$ 만큼 줄어든다. 앞서서도 언급했듯이, 하이폴리곤이 되면  $kd$ -tree의 성능을 만족스럽지 않게 나타난다.

많은 연구자들이 모델 표면 근처의 거리장에 대한 다양한 최적화 자료구조를 제시했다[14,15,16,17]. 그러나 이 방식은 쿼리 포인트가 표면에서 멀리 떨어진 경우 효율적이지 않으며, 하이폴리곤이 되면 계산이 오래 걸린다.

Wu와 Kobbelt는 삼각형 메시에서 양자화된 하이퍼평면(Hyperplane)을 BSP(Binary space partitioning)트리 기반으로 정의하고, 이를 통해 효율적으로 거리장을 만들어냈다[18]. 이 방법은 쿼리의 계산 속도를 높일 수 있지만, 고유값(Eigenvalue)을 계산과정 때문에 비효율적이다.

CDF(Complete distance field)는 정규격자의 각 노드에서 가장 가까운 프리미티브(예 : 삼각형)의 모든 인덱스들을 저장한다[19]. 이 방법은 쿼리 포인트가 영역에 존재하는 노드에 저장된 프리미티브에 대한 모든 거리 값을 계산한다. 일반적으로 CDF는 많은 프리미티브를 탐색해야 하기 때문에 쿼리 성능이 상대적으로 느리다. 이 방법의 옥트리(Octree) 버전은 메모리 비용을 줄이지만[20], 상대적으로 노드당 더 많은 프리미티브가 필요하기 때문에 일반적으로 CDF보다 느리다.

옥트리 구성을 기반으로 샘플링을 적응적으로 수행하는 접근법들도 있다[10,11]. 특히, Lefebvre와 Hoppe는 데이터 저장의 효율성을 향상시키기 위해 컴팩트 데이터 구조를 새롭게 제안했다[11]. Frisken 등의 은 정규격자보다 저장 공간을 덜 사용하지만[10], 보간이 필요하므로 높은 곡률 영역에서 왜곡이 발생한다 (Fig. 2a 참조).

많은 GPU 기반 방법들은 거리값을 정규격자에 저장한다[21,22,23]. 결과적으로 이 방식에 대한 저장 공간과 계산 시간은 격자 해상도인  $l$ 에 따라  $O(l^3)$ 가 된다. Yin과 Liu는 적응적으로 샘플링된 거리장을[24] 구성하는 GPU 기반 방법을 제안하였다[10].

최근에 1차 일반화(First-order generation) 기법을 활용하여 부호 거리장의 정확도가 저장 효율성을 향상시킬 수 있는 방법을 제안되었다[25]. 이들은 거리 함수의 공간

도함수(Spatial derivatives)를 거리 필드 샘플에 통합하는 새로운 방식을 제안하였다.

## III. The Proposed Scheme

### 1.1 CUDA-Based Parallel Signed Distance Field

여러 샘플링 포인트로 구성된 3D 메시에 대한 부호 거리장을 계산하는 것은 1:N관계로 간주될 수 있으므로, 병렬화 구조에 쉽게 적용이 가능하다. 여기서 1:N관계는 하나의 샘플링 포인트에서 거리 값을 결정하기 위해서는 여러 개의 삼각형을 고려해야 되기 때문이다 (Fig. 3 참조).

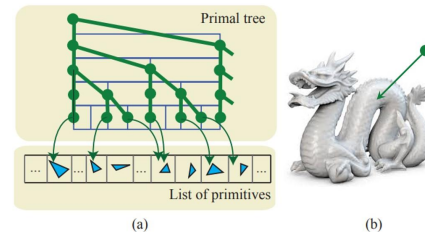


Fig. 3. Adaptive SDF structure : (a) 1D primal tree, with each node containing an index to the nearest primitives of (b) a composite object.

본 논문의 주안점은 런타임 동안 점진적으로 생성되는 적응형 샘플링 격자에서 부호 거리장을 생성하는 것이다. 메시의 삼각형 개수가 하이폴리곤일수록 GPU에서 제공하는 스레드 개수보다 많아 질 수 있다. 본 논문에서는 GPU의 메모리 대역폭을 효율적으로 활용하기 위해  $k$ -d tree를 GPU 메모리에 효율적으로 복사하여 활용하는 접근법을 사용한다.

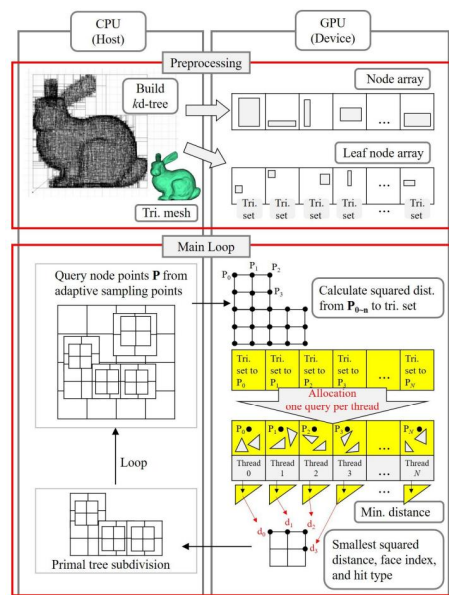


Fig. 4. Algorithm overview.

GPU는 CPU에서 사용하는 트리구조를 효과적으로 사용할 수 없으며, 빌드 또한 비효율적이다. 따라서 CPU에서 트리를 빌드하고 GPU에서 효과적으로 사용할 수 있도록 데이터를 변환시켜 GPU에 복사한다. GPU의 메모리는 CPU에 비해 용량이 작고 트랜잭션의 크기를 줄일수록 가속화되기 때문에 최소한의 데이터만을 GPU로 복사하는 것이 효율성을 극대화할 수 있다.

Fig. 4는 본 논문에서 제안하는 방법의 알고리즘 개요를 보여준다. 전처리 과정에서  $k$ -d tree는 CPU환경에서 계산되고, 구성된 트리의 각 노드는 일반노드와 리프노드로 분류되어 GPU 메모리로 전달한다.

## 2. CUDA-Based Construction of $k$ -d Tree

우선  $k$ -d tree를 활용하여 쿼리 지점  $q$ 에서 거리를 계산하는 방법에 대해 설명한다. 지점  $q$ 에서 가장 가까운 리프노드를 찾고, 그 노드 내에 정점들 중 가장 가까운 거리를 계산한다. 이때 일반노드와 리프노드는 필요한 데이터가 다르다 (Fig. 5 참조).

```

struct Node {
    double divPos;
    unsigned int divAxis : 2;
    unsigned int isLLeaf : 1;
    unsigned int isRLeaf : 1;
    unsigned int left : 30;
    unsigned int right : 30;
};

struct Leaf {
    double min[3];
    double max[3];
    unsigned int numOfPolygon;
    unsigned int polygonIndex;
};

```

Fig. 5. Node structure.

일반노드는 임의의 점에서 가장 가까운 리프노드를 찾는 데 필요로 하는 삼각형을 나누는 축, 축의 위치, 자식노드의 위치 데이터만을 필요로 하며, 리프노드는 삼각형, 그리고 노드의 범위 데이터만 필요하다. 따라서 일반노드와 리프노드를 구분하여 GPU로 복사한다 (Fig. 4의 전처리 과정). 일반노드에 해당하는 Node의 divAxis는 나누는 축이 X, Y, Z임을 구분하는 데이터로 2비트의 크기로 충분히 계산할 수 있다. 또한, divPos는 축의 위치로 double형식이고, isLLeaf와 isRLeaf는 왼쪽과 오른쪽 자식노드가 리프노드인지를 구분하기 때문에 1비트로 충분하다. 남은 left와 right는 왼쪽 자식노드와 오른쪽 자식노드의 인덱스로 노드의 크기를 최소한으로 만들기 위해 16바이트가 되도록 30비트로 설정한다.

리프노드는 노드의 범위 데이터를 min, max로 저장했고, 삼각형 데이터는 따로 저장한 후 노드에 해당하는 삼각형의 시작 인덱스와 개수를 데이터로 가진다. 자식노드 및 삼각형 데이터를 인덱스로 저장하는 이유는 GPU에서 1차원 데이터 형식이 효율적이기 때문에 노드 및 삼각형 데이터를 배열로 저장한다.

트리를 복사하는 과정은 다음과 같다. GPU로 복사할 데이터는 모두 배열로 저장 : 일반노드, 리프노드, 그리고 삼각형 집합. CPU 트리의 루트노드로부터 너비 우선 탐색해 가며 일반노드면 Node형태로 복사하여 배열에 순차적으로 저장한다. 이때 노드의 자식노드가 일반노드면 일반노드 인덱스로, 리프노드면 리프노드 인덱스로 저장한다. 탐색된 노드가 리프노드라면 Leaf형태로 복사하여 리프노드 배열에 순차적으로 저장한다. 동시에 노드의 삼각형들을 복사하고, 첫 삼각형의 인덱스와 삼각형의 개수를 노드에 저장하면 전처리 과정은 완료된다.

## 3. CUDA-Based Construction of Primal Tree

앞에서 계산한  $k$ -d tree를 사용하여 Primal tree를 빌드 할 때 기존 방법으로 빌드를 하면 효율성이 떨어지기 때문에 본 논문에서는 새로운 최적화 방법을 제시한다. GPU기반  $k$ -d tree는 동시에 계산하는 점들이 많을수록 효율이 높아지는 특징이 있기 때문에 본 논문에서는 GPU기반  $k$ -d tree로 동시에 많은 양을 계산할 수 있도록 빌드 방법을 디자인한다.

Fig. 4에서 호스트 CPU는 Primal tree 공간 구조를 기반으로 공간을 분할하고, GPU는 각 스레드에서 계산해야 될 쿼리 노드의 위치인  $P_{0,\dots,N}$ 를 전달한다. 쿼리 위치  $P$ 는 모든 스레드에 할당된다. 각 스레드는  $P$ 에서 포함된 삼각형까지의 거리를 계산한다.

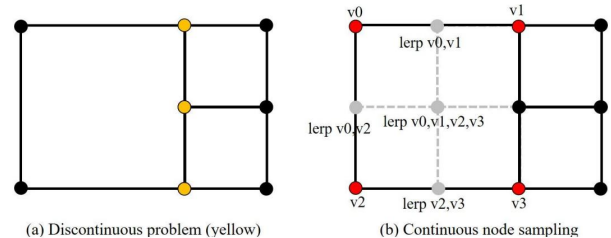


Fig. 6. Node sampling strategy in primal tree.

노드를 탐색할 때 같은 깊이의 노드들을 모두 Queue에 저장하여 관리하고, 저장된 노드들의 자식노드들을 GPU기반  $k$ -d tree로 한 번에 샘플링한다. 예를 들어, 3번째와 4번째, 6번째 자식노드가 분할된다면 해당 subDiv는

00110100이 된다. Primal tree에서 노드를 샘플링하는 방법은 Fig. 6과 같다. 불연속이 나타나는 노란색 노드에 대해서 주변 값을 보간하여 한번 더 샘플링함으로써 불연속 문제를 최소화한다.

분할 과정이 끝나고 노드들의 오차 값을 계산하는데, subDiv값이 0이 아닌 노드만 오차 값을 계산하고, 임계 값 미만이면 subDiv값을 0으로 설정한다. 이때 오차 값 계산 또한 GPU기반  $k$ -d tree를 이용하여 계산한다. 마지막으로 Queue를 비워주고, 0이 아닌 부분에 해당하는 자식노드들만을 생성하여 Queue에 저장한다. Queue에 저장된 노드들의 깊이가 목표 레벨에 도달했거나, 저장된 노드가 없을 때 까지 이를 반복한다.

#### IV. Results

본 연구의 결과들을 만들기 위해 실험한 환경은 Intel Core i7-7700K CPU, 32GB RAM, Geforce GTX 1080Ti GPU가 탑재된 컴퓨터를 이용하였다. 본 연구에서는 CUDA를 이용하여 적응형 부호 거리장을 연속적인 스칼라 필드 형태로 표현하였다. 제안한 방법의 효율성을 정확성을 실험하기 위해 3가지 시나리오를 통해 검증을 했다.

첫 번째로 적응형 부호 거리장의 절단면을 시각화하여 연속성과 적응형 노드 분할 구조를 시각화한다.

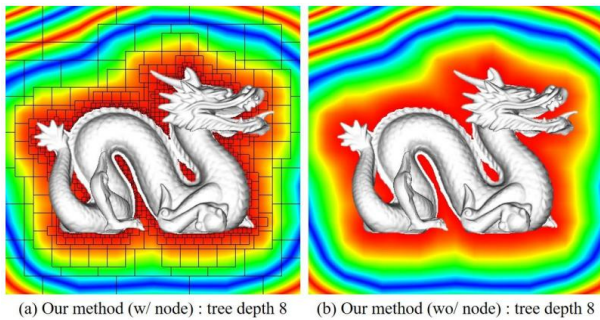


Fig. 7. Adaptive signed distance field with our method (input mode : Stanford dragon, tree depth : 8, error tolerance :  $6.25 \times 10^{-5}$ ).

Fig. 7~8에서 보듯이 연속적인 거리 값의 변화가 나타났으며, 이 같은 결과를 Fig. 1a와 비교했을 때 둘 다 적응형 트리 구조임에도 불구하고 불연속적인 문제가 월등히 줄어들었음을 잘 보여준다. 적응형 부호 거리장을 구축하는데 걸리는 시간에 대해서 우리의 방법과 CPU로 구현한 방법을 비교한 결과, 우리의 방법은 13배정도 빠른 결과를 보였다 (Table 1 참조).

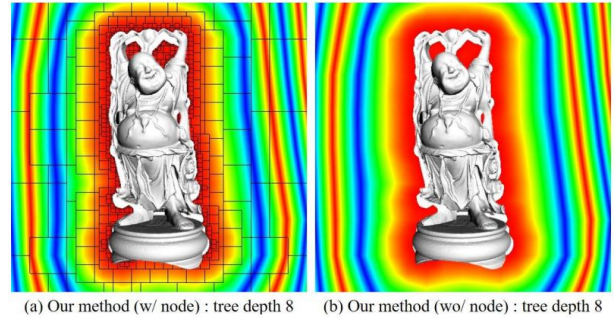


Fig. 8. Adaptive signed distance field with our method (input mode : Stanford happy buddha, tree depth : 8, error tolerance :  $6.25 \times 10^{-5}$ ).

Table 1. Performance

Model	# of tri.	$k$ -d tree (CPU)	$k$ -d tree (GPU)
Bunny	1,500	1.3 sec	0.29 sec
Bunny	69,666	12.01 sec	1.19 sec
Dragon	100,000	11.27 sec	1.24 sec
Buddha	100,000	12.19 sec	1.24 sec

또한, 레이트레이싱 기법을 통해 우리 기법의 표면 정확성을 실행했으며 (Fig. 9 참조), 표면 노이즈 없이 깔끔하게 원본 메쉬 형태를 잘 표현했다. 이 결과에서 CPU는 프레임당 0.5초가 소요되었지만, 우리의 방법은 프레임당 0.025초가 소요되었다. CPU는 실시간 애플리케이션에서는 활용하기에 충분하지 않은 결과를 보였지만, 제안하는 방법은 실시간 애플리케이션에서 적용할 수 있을 만큼의 FPS가 측정되었다.

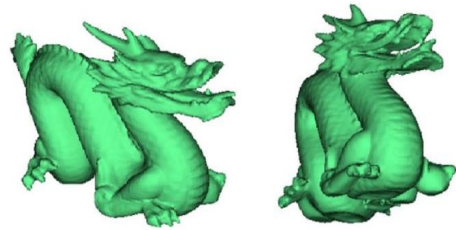


Fig. 9. Real-time raytracing with sphere tracing method[26].

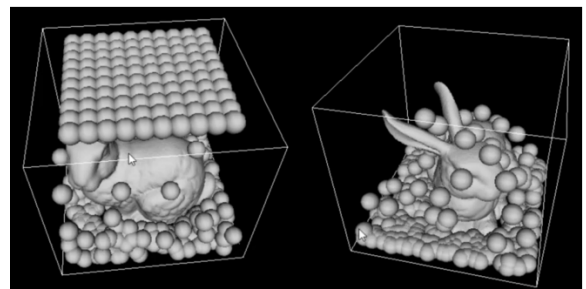


Fig. 10. Real-time collision handling[x].

Fig. 10은 본 논문에서 제안하는 방법을 이용하여 입자 충돌처리에 적용한 결과이다. 적응형 부호 거리장이 전처리 가정으로 계산되기 때문에 실시간 충돌 검출 및 처리가 가능하며, 충돌처리는 레벨셋 기반의 접근법을 이용하였다 [5]. 충돌 후의 속도는 아래 수식과 같으며 좀 더 자세한 설명은 Bridson 등의 논문을 읽어보길 추천한다[5].

$$v_{T,rel}^{new} = \max\left(\frac{v_{p,N}^{new} - v_{p,N}}{|v_{T,rel}|}\right)v_{T,rel}$$

여기서  $\mu$ 는 마찰력 계수이다. 최종적인 충돌 후의 속도는 다음과 같다 :  $v_{p,T}^{new} = v_T + v_{T,rel}^{new}$ . 뿐만 아니라, 부호 거리장을 구축하는데 있어서 매 프레임 약 1초 내외로 계산되어 지기 때문에 강체가 아닌 변형체 시뮬레이션에서도 얼마든지 활용이 가능하다.

## V. Conclusions

본 논문에서는 GPU 기반으로 적응형 및 연속적인 부호 거리장을 최적화하여 구성, 시각화, 그리고 이를 활용한 실시간 레이트레이싱 및 충돌처리 결과를 보여주었다. 향후 볼륨 기반 필드가 아닌 삼각형 프리미티브를 이용하여 좀 더 정확한 결과를 만들 수 있을 방법으로 확장할 예정이다. 뿐만 아니라, 적응형 부호 거리장을 이용하여 변형체의 자기충돌(Self-collision) 처리까지 실시간으로 처리할 수 있는 최적화 기법에 대해서 연구할 계획이다.

## REFERENCES

- [1] Losasso, F., Talton, J., Kwatra, N. and Fedkiw, R., "Two-way coupled SPH and particle level set fluid simulation". IEEE Transactions on Visualization and Computer Graphics, Vol. 14, No. 4, pp.797-804, 2008. DOI: 10.1109/TVCG.2008.37
- [2] Hong, W. and Wang, Y., "A coupled level set and volume-of-fluid simulation for heat transfer of the double droplet impact on a spherical liquid film". Numerical Heat Transfer, Part B: Fundamentals, Vol. 71, No. 4, pp.359-371, 2007. DOI: 10.1080/10407790.2017.1293960
- [3] Mullen, P., McKenzie, A., Tong, Y. and Desbrun, M., "A variational approach to Eulerian geometry processing". ACM Transactions on Graphics (TOG), Vol. 26, No. 3, pp.66, 2007. DOI: 10.1145/1275808.1276459
- [4] Osher, S. and Paragios, N. eds., "Geometric level set methods in imaging, vision, and graphics". Springer Science & Business Media, 2003.
- [5] Bridson, Robert, Sebastian Marino, and Ronald Fedkiw. "Simulation of clothing with folds and wrinkles." In ACM SIGGRAPH Courses, pp. 3, 2005.
- [6] Selle, A., Su, J., Irving, G. and Fedkiw, R., "Robust high-resolution cloth using parallelism, history-based collisions, and accurate friction". IEEE transactions on visualization and computer graphics, Vol. 15, No. 2, pp.339-350, 2008. DOI: 10.1109/TVCG.2008.79
- [7] Westermann, Rüdiger, Christopher Johnson, and Thomas Ertl. "A level-set method for flow visualization." In Proceedings IEEE Visualization, pp. 147-154. 2000. DOI: 10.1109/VISUAL.2000.885688
- [8] Klemelä, J., "Visualization of multivariate density estimates with level set trees". Journal of Computational and Graphical Statistics, Vol. 13, No. 3, pp.599-620, 2004. DOI: 10.1198/106186004X2642
- [9] Horn, Daniel Reiter, Jeremy Sugerman, Mike Houston, and Pat Hanrahan. "Interactive kd tree GPU raytracing." In Proceedings of the symposium on Interactive 3D graphics and games, pp. 167-174. 2007. DOI: 10.1145/1230100.1230129
- [10] Frisken, Sarah F., Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. "Adaptively sampled distance fields: A general representation of shape for computer graphics." In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 249-254. 2000. DOI: 10.1145/344779.344899
- [11] Lefebvre, Sylvain, and Hugues Hoppe. "Compressed random-access trees for spatially coherent data." In Eurographics Symposium on Rendering Techniques, pp. 339-349. Eurographics Association, 2007.
- [12] Luke, H.D., "The origins of the sampling theorem". IEEE Communications Magazine, Vol. 37, No. 4, pp.106-108, 1999. DOI: 10.1109/35.755459
- [13] Park, Taejung, Sung-Ho Lee, Jong-Hyun Kim, and Chang-Hun Kim. "Cuda-based signed distance field calculation for adaptive grids." IEEE International Conference on Computer and Information Technology, pp. 1202-1206, 2010. DOI: 10.1109/CIT.2010.217
- [14] Mauch, Sean Patrick. "Efficient algorithms for solving static Hamilton-Jacobi equations". California Institute of Technology, 2003.
- [15] Nielsen, M.B. and Museth, K., "Dynamic Tubular Grid: An efficient data structure and algorithms for high resolution level sets". Journal of Scientific Computing, Vol. 26, No. 3, pp.261-299, 2006. DOI: 10.1007/s10915-005-9062-8
- [16] Houston, B., Nielsen, M.B., Batty, C., Nilsson, O. and Museth, K., "Hierarchical RLE level set: A compact and versatile

- deformable surface representation". ACM Transactions on Graphics (TOG), Vol. 25, No. 1, pp.151-175, 2006. DOI: 10.1145/1122501.1122508
- [17] Chang, Byungjoon, Deukhyun Cha, and Insung Ihm. "Computing local signed distance fields for large polygonal models." In Computer Graphics Forum, Vol. 27, No. 3, pp. 799-806, 2008. DOI: 10.1111/j.1467-8659.2008.01210.x
- [18] Wu, Jianhua, and Leif Kobbelt. "Piecewise Linear Approximation of Signed Distance Fields." In VMV, pp. 513-520. 2003.
- [19] Huang, Jian, Yan Li, Roger Crawfis, Shao-Chiung Lu, and Shuh-Yuan Liou. "A complete distance field representation." In Proceedings IEEE Visualization, 2001. VIS'01., pp. 247-561, 2001. DOI: 10.1109/VISUAL.2001.964518
- [20] Huang, Jian, and Roger Crawfis. "Adaptively represented complete distance fields." Geometric modeling for scientific visualization, 2002.
- [21] Sud A, Otaduy MA, Manocha D. "DiFi: Fast 3D distance field computation using graphics hardware". Computer Graphics Forum, Vol. 23, No. 3, pp. 557-566, 2004. DOI: 10.1111/j.1467-8659.2004.00787.x
- [22] Qu H, Zhang N, Shao R, Kaufman A, Mueller K. "Feature preserving distance fields". IEEE Symposium on Volume Visualization and Graphics, pp. 39-46, 2004. DOI: 10.1109/SVVG.2004.3
- [23] Sud A, Govindaraju N, Gayle R, Manocha D. "Interactive 3d distance field computation using linear factorization". In Proceedings of the 2006 symposium on Interactive 3D graphics and games, pp. 117-124, 2006. DOI: 10.1145/1111411.1111432
- [24] Yin, K., Liu, Y. and Wu, E., "Fast Computing Adaptively Sampled Distance Field on GPU", Pacific Graphics Short Papers, 2011. DOI: 10.2312/PE/PG/PG2011short/025-030
- [25] Bán R, Valasek G. "First Order Signed Distance Fields", In Eurographics Short Papers, pp. 33-36, 2020.
- [26] Hart, John C. "Sphere tracing: Simple robust antialiased rendering of distance-based implicit surfaces." In SIGGRAPH, vol. 93, pp. 1-11. 1993. DOI: 10.1007/s003710050084

## Authors



Jong-Hyun Kim received the B.A. degree in the department of digital contents at Sejong University in 2008. He received M.S. and Ph.D. degrees in the department of computer science and engineering at Korea University,

in 2010 and 2016. Prof. Kim is an associate professor in the school of software application in Kangnam University. His current research interests include fluid animation and virtual reality.