

GPU 기반의 특정 영역에 대한 파편 낙하 계산 모델

김태권, 조규태, 이승영
LIG넥스원(주)

e-mail:taegwon.kim@lignex1.com, kyutae.cho@lignex1.com, seungyoung.lee@lignex1.com

GPU based Shrapnel Drop Computational Model for Specific Area

Tae-Gwon Kim, Kyu-Tae Cho, Seung-Young Lee
LIG Nex1 Inc.

요 약

특정 영역에 낙하하는 파편에 대한 계산은 파편의 개수가 증가할수록 계산량이 급격히 늘어나기 때문에 많은 자원이 소비된다. 이러한 파편의 낙하 계산은 각각의 파편이 서로 영향을 받지 않기 때문에 일반적으로 CPU나 GPU를 활용하여 병렬로 연산을 수행할 수 있다. 이 논문에서는 특정 영역에 낙하하는 파편을 효율적으로 계산하기 위한 GPU 기반의 파편 낙하 계산 설계 모델을 제안한다. 이 모델은 공중의 특정점에서 폭발한 물체의 파편 방향을 계산한 후, 해당 방향으로 이동한 각각의 파편들이 떨어지는 방향에 대해 트리형식으로 계산을 반복적으로 수행해 최종 낙하 위치를 도출한다. 제안하는 방법은 GPU를 활용하여 파편의 낙하 영역을 사진트리를 통해 하향식(top-down)으로 계산하므로 넓은 영역에 대해 효율적으로 낙하점을 계산할 수 있다.

1. 서론

최근 사용자들이 유도탄 폭발로 발생하는 파편의 낙하, 우주 파편의 낙하 등, 특정 영역에서의 종전보다 상세한 파편 낙하 시뮬레이션을 요구하면서 정확한 시뮬레이션을 위해 넓은 영역에 많은 수의 파편을 사용하게 되었다. 하지만 파편들의 낙하점 계산은 하드웨어의 발전에도 불구하고 늘어나는 파편의 숫자와 넓어지는 계산 영역으로 인해 많은 시간을 소모한다. 따라서 파편들의 운동을 효율적으로 계산할 수 있는 모델에 대한 연구[1][2]들이 진행되고 있다.

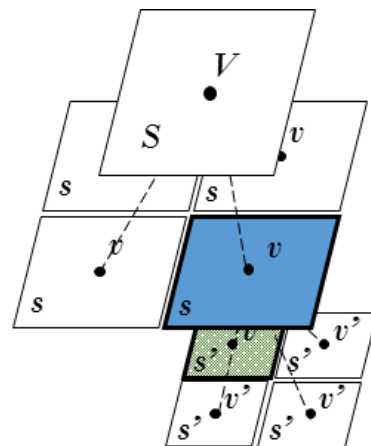
이 논문에서는 지형렌더링 분야에서 사용되는 기하 분할(geometry splitting)[3] 기법을 활용하여 영역 설정을 위한 사진트리를 구성한다. 기하 분할 기법은 기존의 CPU 기반의 사진트리 삼각화 기법의 문제점을 해결하기 위해 고안된 기법으로 기하 셰이더(geometry shader)의 스트림 출력 단계(stream output stage)를 이용해 상세단계를 선별 하면서 정점들을 재귀적으로 증식시켜 사진트리의 탐색을 수행하므로 GPU를 활용하여 트리를 탐색할 수 있는 기법이다.

제안하는 모델에서는 기하 분할 기법을 응용하여 기하 분할을 상세단계 선별(LOD: Level of Detail)이 아닌 특정 영역에서의 파편들의 낙하 영역을 계산하는데 사용한다. 파편들이 이동하는 방향에 대한 근사 영역들을 분할된 정점에 할당하여 사진 트리를 구성하고 해당 영역에서의 파편의 이동 연산을 반복적으로 수행하면서 최종 낙하지점을 결정한다.

이 방법은 GPU 내부에서 재귀연산을 통해 트리를 구성하므로 GPU를 활용하여 낙하 영역을 구성할 수 있고 파편의 이동방향 연산이 트리가 깊어질수록 점점 더 병렬적으로 수행되므로 독립적인 다수의 파편에 대해 효율적인 연산이 가능하다.

2. GPU 기반의 파편 낙하 계산 모델 설계

이 논문에서는 기하 분할 기법의 장점을 응용하고 GPU의 병렬 연산을 이용하여 특정 영역에서 낙하하는 파편에 대한 연산을 효율적으로 수행하는 모델을 제안한다.

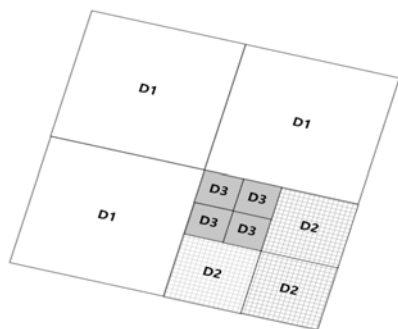


(그림 1) 최초 파편 발생 영역 S의 낙하점 V로부터 V에서 분할된 정점 v의 영역 s를 반복적으로 설정하는 예시

[그림 1]은 제안하는 방법을 통해 파편 낙하 발생 지점으로 부터 낙하 종료 시점까지의 계산 모델을 간략화한 그림이다. 모델은 우선 파편이 처음으로 발생하는 지역 S의 낙하 시작점인 정점 V를 부모 노드로 지정하고 초기 낙하 영역인 S에서 낙하에 사용되는 식인 e를 적용하여 파편이 이동하는 근사 영역 s의 정점인 v에 이동한 파편들을 할당하여 설계한다. 여기서 영역 S는 시뮬레이션을 수행하고자 하는 전체 영역이고 정점 v는 정점 V로부터 분할되어 트리를 구성하는 정점이다. 그리고 영역 s는 전체 영역인 S를 4분할한 영역이며 식 e는 제안하는 낙하 모델에 적용 가능한 모든 식이다.

이 후, 낙하점으로부터 분할된 4개의 정점 v의 영역인 s가 생성되면 해당 영역에 속하는 파편에 대해 식 e를 적용해 다음 이동 영역인 s'을 구해가는 과정을 파편이 최종 낙하지점에 도달할 때 까지 반복한다. 정점이 분할된 후 각 단계에서는 영역 s에 속하는 파편들의 이동 방향에 대한 계산을 수행하게 된다. 이 모델은 전체 영역 S를 사진트리를 활용하여 점점 세세한 영역인 s로 나뉘가며 최종 낙하점을 찾으므로 연산을 반복할수록 근사 영역 안의 파편들은 근사 영역이 아닌 실제 낙하 영역에 수렴하게 된다.

제안하는 모델은 영역을 구성할 때 사진트리를 이용해 구성하므로 이 과정을 n회 수행할 때마다 4ⁿ개의 영역을 가지게 되며 4ⁿ개의 영역에 속하는 파편들을 GPU를 활용해 병렬적으로 연산을 수행하므로 파편이 낙하지점에 도달할수록 높은 정확성과 병렬성을 가지게 된다. DirectX 11 기반의 단일 텍스처로 이루어진 영역에 대해 모델을 적용할 경우 최대 8192×8192의 크기(67,108,864개의 영역)를 가지는 영역 S에 대해 연산을 수행할 수 있으며 단일 텍스처로 이루어지지 않은 임의의 영역에 대해서는 하드웨어나 소프트웨어의 한계까지 연산 수행이 가능하다.



(그림 2) 파편이 속한 영역에 따른 계산의 깊이

사진트리로 구성된 영역들을 계산할 때 트리의 깊이를 낮춰 계산의 오버헤드를 줄이기 위해 파편이 도달하지 않는 영역의 정점에 대해서는 정점분할을 수행하지 않으며 해당연산의 깊이는 트리의 깊이와 같다. [그림 2]는 [그림 1]에서의 영역 s에서부터 정점이 분할되었을 때 연산의 깊이가 Dn을 나타낸 것이며 s'의 다음 영역이 최종 낙하지점일

때의 연산의 깊이 n를 나타낸다. 그림에서 D1은 파편이 도달하지 않은 영역이며 D2는 파편이 1회 도달한 영역, D3는 파편이 2회 도달한 영역이다.

3. 결론 및 향후 연구

이 논문에서는 기하 분할 기법을 응용하여 분할되는 정점에 대해 영역을 할당한 후 파편의 낙하 운동을 반복적으로 계산하여 파편의 최종 낙하지점을 병렬적으로 계산할 수 있는 모델을 제안하였다. 이는 폭발되는 탄두 파편의 낙하와 같은 공중에서 폭발로 인해 발생한 다수 파편의 낙하지점 예측을 시뮬레이션 하는데 효율적으로 사용될 수 있다. 제안하는 방법은 사진트리의 규모가 커지고 트리가 깊어질수록 병렬성을 가지기 때문에 파편들이 넓은 영역에 대해 다수의 방향으로 낙하할 때 효율성이 좋으나 단일 방향으로 이동하는 파편 다발에 대해서는 한계점을 가진다.

향후 연구에서는 이런 한계점을 극복하기 위해 본 모델에 계산 셰이더나 CUDA 등의 GPGPU를 위한 API를 활용해 단일 방향으로 이동하는 파편 다발을 병렬적으로 처리할 수 있는 방안을 적용하여 실제 탄두의 폭발 모델에 대한 시뮬레이션을 수행하는 방법에 대한 연구를 진행할 예정이다.

참고문헌

[1] Jong-wan Lee, Chang-geun Song, SangYoon Yoon. "Modeling of Fragments of Explosion considering Internal Force of Particle System." Journal of The Korean Society for Computer Game, Vol.25, No.2 pp.17-23, 2012.

[2] Sung-Jang Wang, Chang-Hun Kim, "A Large Number of Fragments Control Plug-in Framework for Maya", Proceedings of the Korea Information Processing Society Conference, Vol.17, No.02 pp.795-798, 2010.

[3] E.S Lee, B.S Shin, "Geometry splitting: an acceleration technique of quadtree-based terrain rendering using GPU," IEICE TRANSACTIONS on Information and Systems, Vol.E94-D, No.1, pp.137-145, 2011.