본 논문에서는 GPU기반 대량의 파티클 간의 실시간 충돌처리 알고리즘을 제안한다. 해당 연구 프로젝트는 OpenGL로 진행하였고 3차원 파티클 위치 정보를 2차원 이미지에 투영하고 이를 기반으로 충돌 처리를 수행하는 단계와 실제 렌더링을 수행하는 두단계를 거친다.

첫 단계에선 파티클간의 충돌처리를 수행하며 위치를 업데이트한다. 이때 충돌처리에 사용되는 데이터는 파티클의 3차원 월드 공간의 위치를 컬러 코드로 저장한 2차원 형태의 텍스처이며 이를 충돌 감지 텍스처라 칭한다.

해당 단계의 첫 프레임에서는 CPU를 통해 정점 데이터(Vertex Data)를 입력으로 받으며 업데이트를 진행한다. 이때 충돌 감지 텍스처가 생성되지 않았기 때문에 충돌처리가 진행되지 않고 파티클의 업데이트만 진행한다.

충돌 감지 텍스처 생성 이후 프레임에서는 각 파티클의 충돌 처리를 위해 입력된 정점 데이터에 뷰 프로젝션 행렬(View Projection Matrix)를 곱하여 -1부터 1사이의 값을 가지는 화면 공간 좌표(Screen Space Coordinate)로 변경한다. 변경된 데이터의 x, y좌표를 변형하여 0부터 1사이의 값을 가지는 텍스처 좌표(Texture Coordinates)로 변환하고 해당 좌표 값에 충돌 감지 텍스처의 가로, 세로 해상도 크기를 곱하여 현재 입력된 파티클의 텍셀(Texel)을 얻는다. 이 정보를 이용해 texelFetch 함수를 사용하여 충돌 감지 텍스처에 저장된 3차원 위치 정보를 읽어온다. 주어진 탐색 범위를 샘플링하며 해당 범위 내에 파티클의 3차원 위치 정보를 얻어 현재의 파티클 위치와의 거리를 계산하여 비교한다. 그중 가장 작은 값과 충돌을 진행한다.

첫 단계의 기하 셰이더(Geometry Shader)에서는 Transform Feedback 기능을 통해 파티클 정점의 데이터를 Transform Feedback Buffer Object에 갱신하며, 프래그먼트 셰이더(Fragment Shader)에서 파티클의 3차원 좌표 형태의 값을 컬러 값으로 작성하여 충돌 탐지 텍스처를 갱신한다.

두번째 단계는 첫번째 단계에서 업데이트된 Transform Feedback Buffer Object를 입력 데이터로 이용하여 기하 셰이더(Geometry Shader)를 통해 빌보드 형태의 쿼드로 변형하여 파티클을 그려주는 단계를 거친다.