

장애물 객체의 회전 벡터를 이용한 VR 환경에서의 효율적인 음향 처리 및 합성

박성아^o, 박소연*, 김종현*

^o강남대학교 소프트웨어응용학부,

*강남대학교 소프트웨어응용학부

e-mail: jonghyunkim@kangnam.ac.kr

Efficient Sound Processing and Synthesis in VR Environment Using Curl Vector of Obstacle Object

Seong-A Park^o, Soyeon Park*, Jong-Hyun Kim*

^oSchool of Software Application, Kangnam University,

*School of Software Application, Kangnam University

● 요약 ●

본 논문에서는 장애물 객체의 회전 벡터를 이용하여 VR 환경에서의 효율적으로 음향 처리 및 합성하는 방법을 제안한다. 현실에서 소리와 장애물이 있을 때, 소리는 장애물의 형태에 따라 퍼지면서 전파되는 형태를 보여준다. 이 같은 특징을 가상현실 환경에 유사하게 음향 처리하고자 하며 이를 위해 장애물 객체의 위치와 소리의 근원지 위치를 입력으로 소리의 전파 형태를 근사한다. 이때 모서리 부근에서 표현되는 소리의 회전을 계산하기 위해 장애물의 회전벡터(Curl vector)를 기반으로 소리의 회전을 추출하였으며, 장애물 형태를 컨볼루션(Convolution)하여 소리가 바깥 방향으로 전파되는 형태를 모델링한다. 또한, 장애물과 소리 벡터 사이의 거리, 소리 근원지와 소리 벡터 사이의 거리를 계산하여 소리의 크기를 감쇠 시켜 주며, 최종적으로 장애물 주변으로 퍼지는 벡터 모양인 외부벡터를 합성하여 장애물로부터 외부로 퍼지는 벡터의 방향을 설정한다. 본 논문에서 제안하는 방법을 이용한 소리는 장애물과의 거리와 형태를 고려하여 퍼지는 사운드 벡터 형태를 보여주며, 소리 위치에 따라 소리 감소 패턴이 변경되고, 장애물 모양에 따라 흐름이 조절되는 결과를 보여준다. 이 같은 실험은 실제 현실에서 소리가 장애물의 모양에 따라 나타나는 소리의 변화 및 패턴을 거의 유사하게 표현할 수 있다.

키워드: 소리(Sound), 장애물(Obstacle), 회전벡터(Curl vector), 가상환경(Virtual environment), 소리 확산(Sound diffusion)

I. Introduction

현재 메타버스와 같은 가상공간에 관한 관심이 집중되면서 가상공간에서 현실감을 극대화할 수 있는 기술이 중요해지고 있다. 현재 이러한 기술은 현실감을 느끼게 하는 사람의 오감을 가상환경에서 구현하는 데 중점을 두고 있다. 그중 청각을 구현해 실제 환경과 같은 느낌을 주는 방법들이 많이 제안되었지만[1], 현재까지의 소리는 주로 퍼지기만 하는 소리나, 장애물이 있으면 복잡한 계산을 통해 소리가 전달되는 방법으로 구현되어왔다[2]. 하지만 가상공간에서 단순히 소리가 퍼져 나오는 것은 몰입감을 저해하며 복잡한 계산을 이용하여 실시간 애플리케이션에서 활용이 어려워진다. 본 논문에서는 이 문제를 해결하고자 가상공간에서 객체의 회전 벡터를 이용해

소리를 효율적으로 처리하는 방법을 제안한다.

II. The Proposed Scheme

1. 소리 근원지가 고려된 소리의 회전벡터 계산

본 논문에서는 장애물 객체는 사각형과 같이 각진 형태의 장면을 설정하였고, 이 장면을 기반으로 기본적인 지형과 소리를 다음과 같이 배치한다 (Fig. 1 참조). Fig. 1a에서 파란색 점은 소리의 근원지이며, 장애물의 위치와 소리의 근원지는 사용자가 설정 가능한 값이다.

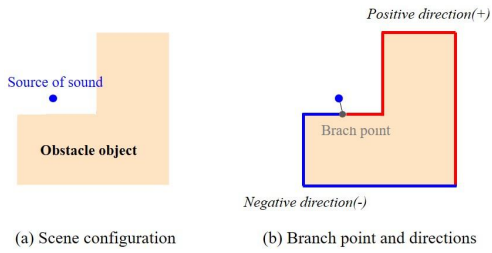


Fig. 1. Basic scene setup.

우선 소리와 가장 가까운 장애물 면을 찾고, 사운드 에너지가 면과의 충돌을 통해 양방향으로 퍼질 수 있는 분기점을 계산한다 (Fig. 1b의 녹색 점 참조). 소리가 퍼질 때, 양의 방향(+)에 해당하는 붉은색 선, 음의 방향(-)에 해당하는 파란색으로 설정한다. 각 선분의 개수는 장애물 정점 n 개의 기점으로 $(n + 1)/2$ 개만큼 나눠 가지게 하였으며, 만약 홀수일 경우 양의 방향이 한 번 더 들게 설정한다 (Fig. 1b 참조).

분기점에 의해 각 면에서 벡터들이 그려져야 하는 방향이 설정되었다면, 면과 선을 기준으로 내/외부 판단이 필요하다. 이를 위해 선분의 중심점을 기준으로 상하좌우 4개를 찍어서 두 정점과 x, y 값 중 같은 좌표를 제외하고 남은 좌표를 이용하여 내부에 있는지 아닌지 판단한다[x]. 최종적으로 이를 통해 각 장애물 면 선을 기준으로 어디가 장애물의 외부임을 알려주는 점을 저장한다 (Fig. 2a 참조). 또한, 장애물의 정점을 이용하여 벡터의 순서를 설정한다. Fig. 2b와 같이 장애물이 있다고 가정할 때, 첫 번째 정점의 인덱스를 1이라 임의로 지정하고 시계방향으로 인덱스를 증가시킨다 (Fig. 2b 참조).

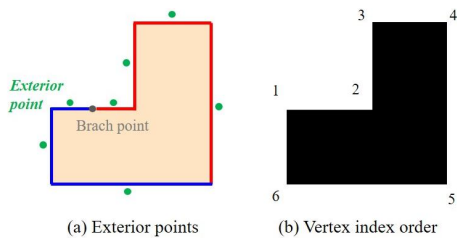


Fig. 2. Exterior points and indexing.

본 논문에서는 회전벡터를 계산할 범위를 설정 할 때, 직선 부분과 회전 부분을 분류한다. 먼저 회전 부분은 현재 점과 이전 점, 그리고 다음 점을 이용하여 회전벡터를 계산할 공간을 판단한다. 예를 들어 4번째 정점에서 계산한다고 가정했을 때 (Fig. 3a), 4번째 정점의 이전 점인 3번째 정점과 다음 점인 5번째 정점의 위치 값을 이용하여 계산한다.

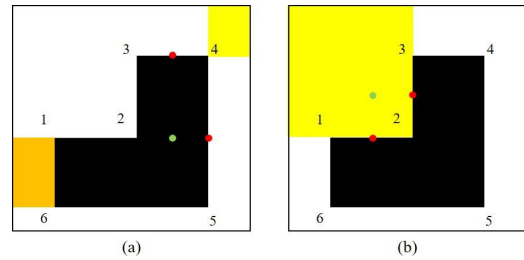


Fig. 3. Area to calculate curl vector.

먼저 4번째와 3번째의 중심점, 그리고 4번째 정점과 5번째 정점의 중심점을 계산한 후, 각 중심점끼리의 중심점을 다시 계산하여 그 점이 장애물의 내부인지 외부인지 판별해 벡터를 그릴 위치를 계산한다. 만약 중심점의 위치가 내부일 경우 Fig. 3a와 같이 외부(노란색 영역)에 벡터를 그리게 하고, Fig. 3b와 같이 외부일 경우 외부(노란색 영역)에 그대로 그려지도록 한다. 그림에서 보여주고 있는 노란색 부분은 회전벡터(Curl vector)가 존재해야 하는 부분이므로 회전벡터를 통해 벡터를 계산한다. 계산방법은 다음과 같다 : 회전 벡터를 그릴 부분의 가장 작은 X 축, Y 축 값과 가장 큰 X 축, Y 축 값을 찾아 배열에 저장하고 양의 방향이면 작은 값에서 큰 값으로 향하는 벡터에 수직이 되는 방향으로 회전 방향을 정해주고, 음의 방향이면 반대 방향으로 방향을 설정한다. 직선 부분은 현재 점과 다음 점 두 점을 이용하여 계산하는데 먼저 두 점의 X 축 또는 Y 축의 정점 중에 어떤 축의 값이 같은지 판별 후 같지 않은 축의 차이를 이용하여 양수인지 음수인지에 따라 방향을 결정한다. 직선 부분은 직선으로 소리가 퍼져나가는 부분으로 좌표만을 이용하여 벡터를 나타내준다 (Fig. 3a의 주황색 영역 참조). 최종적으로, 위 과정을 통해 Fig. 4와 같은 벡터가 만들어진다.

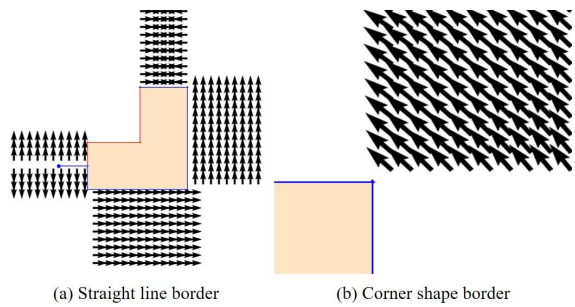


Fig. 4. Vectors in the form of straight lines and corners.

2. 장애물 객체의 컨볼루션 맵을 이용한 효율적인 외부 벡터 계산

본 논문에서는 기본적으로 소리가 일정 거리 떨어질 경우, 감소하는 특징을 고려하여 소리가 들리는 범위를 설정해준다. 이때, 장애물의 외형을 타고 퍼지는 소리를 표현하고자 하므로 장애물의 형태를 고려하여 유한한 범위를 설정한다. 이를 위해 장애물의 형태를 확대해 범위를 지정해주고자 하였고, 컨볼루션 필터링을 통해 다음과 같이 장애물의 형태를 스무딩할 수 있는 맵을 계산한다 (Fig. 5 참조).

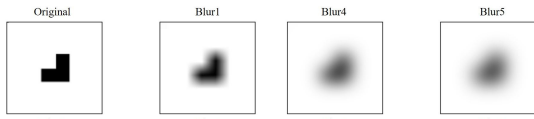


Fig. 5. Smoothing map with convolutional filtering of obstacle object.

스무딩을 적용한 이유는 장애물의 외곽선 부분이 점점 연해지면서 퍼지는 특징이 있는데, 이 특징이 소리가 점차 감소하면서 퍼지는 형태와 유사하므로 본 논문에서는 위 방법을 활용하게 외부 벡터 방향을 계산하였다. 본 논문에서는 가우시안 블러를 7번 적용하여 스무딩 맵을 추출한다 (Fig. 6 참조).

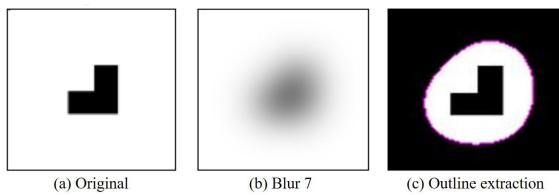


Fig. 6. Extracting outline of obstacle object.

Fig. 6b에서 사용자가 지정한 임계값인 γ 를 두어 스무딩 값이 γ 값보다 큰 경우를 찾아 외곽선에 해당하는 지점인 외곽 점을 추출한다 (Fig. 6c 참조). 외곽 점의 경우 해당 좌표에서 공간 미분을 통해 기울기를 구했으며, 이 값을 기반으로 바깥쪽으로 퍼지는 장애물 객체의 외부벡터를 계산한다. Fig. 7은 최종적으로 소리가 사방으로 퍼지는 벡터인 외부벡터를 보여주는 결과이다 (Fig. 7 참조).

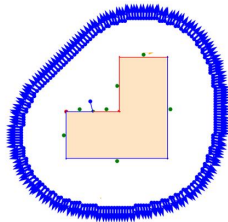


Fig. 7. Exterior vectors.

3. 사운드 벡터 합성

주위에 아무것도 없는 환경에서 소리가 퍼져나가면 등방성하게 퍼지게 되지만, 장애물이 있으면 소리는 장애물의 영향을 받게 되어 흐름이 변하게 된다.

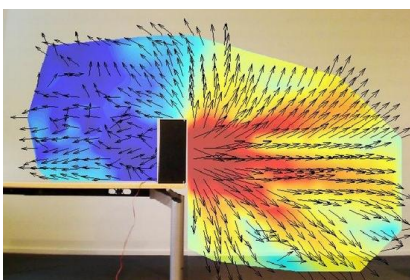


Fig. 8. Visualization of sound field.

이를 표현하기 위해 장애물의 형태와 바깥쪽으로 퍼져나가는 벡터인 외부벡터를 이용하여 소리의 흐름을 합성한다. (Fig. 8 참조). 본 논문에서는 소리의 형태에 따라서만 흐르던 기존 벡터(1장 참조)와 스무딩 맵 기반의 장애물 외부 벡터(2장 참조)를 이용하여 사운드의 흐름을 합성한다. 그리고 벡터의 합성을 위해 장애물 외부에서 가장 가까이 흐르는 벡터를 내부벡터로 지정하여 사용된다. 합성 방법은 현재 위치의 벡터로부터 같은 방향으로 뻗어 나가는 직선을 만들어 일정 거리 나아가도록 그린다. 이후, 직선의 끝점으로부터 가장 가까운 외부벡터를 찾는다. 외부벡터를 찾은 후, 현재 벡터 위치의 벡터로부터 거리상 가장 가까운 내부벡터를 찾는다. 이렇게 두 벡터를 찾은 후 현재 위치로부터 각각의 벡터까지의 거리를 구해준다. 만약 내부벡터가 더 가깝다면 기존에 흐르던 흐름이 더 강하다고 판정되어 기존 벡터에 더 큰 가중치를 주어 외부벡터와 합성되며, 반대로 외부벡터와 더 가깝다면 외부 흐름에 더 영향을 받는다고 판정되어 외부벡터에 더 큰 가중치를 주어 합성된다. 결과적으로 장애물에 가까울수록 장애물 모양과 같은 흐름의 벡터의 형태로 나타내고, 바깥쪽일수록 외부벡터와 같은 사운드 벡터가 나타나게 된다. 이는 Fig. 8에서 보여주는 것과 유사한 형태로 나타난다.

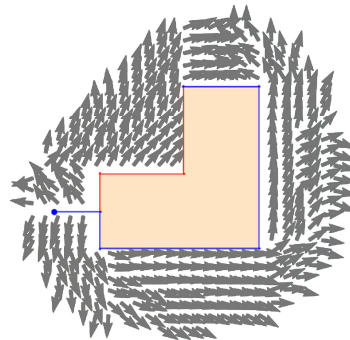


Fig. 9. Sound flow map.

Fig. 9는 위 과정을 통해 외부벡터와 장애물을 고려한 합성 벡터로 장애물에 따른 소리 흐름이 표현된 것을 볼 수 있는 맵이다.

4. 사운드의 흐름 벡터 계산

앞에서 서술한 과정을 통해 장애물의 모양과 거리를 고려한 소리의 방향을 나타낸 소리 흐름 맵을 만들었다면, 이번 장에는 각 위치에 따른 소리의 크기를 화살표의 색으로 표현하여 방향과 강도를 동시에 표현하는 방법에 대해 설명한다. 기본적으로 소리의 크기는 앞에서 언급한 스무딩 필터를 적용하게 되는데 그 이유는 스무딩 맵이 장애물과의 거리를 고려한 감쇠를 표현할 수 있기 때문이다. 결과적으로 소리 근원지와의 거리 가중치 값 α 와 장애물에서부터 떨어진 정도를 나타낸 스무딩 맵에 대한 가중치 값 β 를 합한, $(\alpha + \beta)/2$ 를 통해 평균값을 계산했고, 이를 다시 최댓값으로 나누어 모든 값이 0에서 1 사이 값으로 표현되도록 하였다.

Fig. 10a는 흑백 값을 적용한 결과로 화살표의 색상이 진할수록 소리의 크기 값이 큰 것을 의미한다. Fig. 10b는 컬러를 적용한 결과로 1에 가까울수록 소리가 크다는 것을 의미하여 붉은 계열로

0에 가까울수록 소리가 작다는 것을 의미하며 파란 계열로 화살표의 색상을 표현해준 결과이다. Fig. 10의 결과를 통해 최종적으로 소리 근원지인 파란색 점으로부터 소리가 퍼져나가면서 장애물의 모양을 반영해 흐르는 방향 형태와 소리 근원지와의 거리, 장애물과의 거리를 고려해 감소하는 소리의 크기 형태를 시각적으로 볼 수 있다.

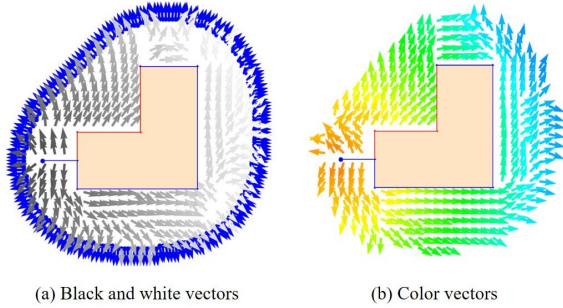


Fig. 10. Visualization of sound field with rainbow color.

4. 가상환경에서의 사운드 실험

앞에서 보여준 결과가 실제로 청각적으로 소리가 장애물 주위에서 잘 계산되어 들리는지 확인해보기 위해 가상환경을 Unity3D로 구성한 장면에서 실험을 진행하였다. 먼저 다음과 같이 소리 근원지는 빨간색 구로 표현해주었다 (Fig. 11 참조). 또한, 사운드의 크기를 색상으로 표현해 시각적으로 소리가 어떻게 들릴지 미리 볼 수 있도록 하였다. 앞에서와 마찬가지로 빨간색에 가까울수록 소리가 크고 파란색에 가까울수록 소리가 작으며, Unity3D의 Listener 객체를 만들어 오브젝트 위에서 움직여보면서 위치에 따라 소리 크기가 변하는 것을 확인할 수 있었다.

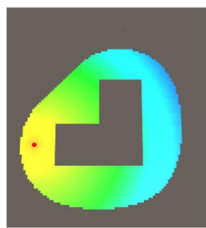


Fig. 11. Result 1.

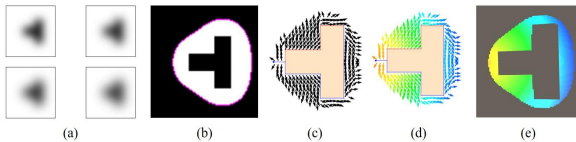


Fig. 12. Result 2 : (a) obstacle smoothing, (b) extracted outline point, (c) sound field, (d) sound field with rainbow color map, (e) sound magnitude in Unity3D.

Fig 12는 다른 모양의 장애물이 있는 환경에서 장애물의 회전벡터를 이용해 소리가 회절되며 확산하는 결과를 보여주는 그림이다. 각 장애물의 모습에 따라 소리의 확산이 잘 이루어지고 있는 것을 확인할 수 있다.

III. Conclusions

본 논문에서는 회전벡터를 이용해 장애물 객체가 존재하는 가상환경에서 실시간으로 소리의 크기를 제어하는 방법을 제시하였다. 이를 통해 장애물이 있는지 지형에서 실제 소리 흐름과 유사하게 표현되도록 퍼져나가는 소리의 흐름을 설계하였다. 또한, 소리의 흐름을 직관적으로 보여주기 위해 소리 흐름 맵을 만들었으며 장애물이 존재하는 상황에서 소리의 흐름과 방향이 잘 표현되었다. 이를 통해 소리뿐만 아니라, 소리에 따른 객체의 이동이나 이유를 표현할 수 있을 거라고 기대한다. 향후, 이 소리에 따라 새와 같은 동물의 이동을 표현하는 방법에 관해 연구할 계획이다.

REFERENCES

- [1] Begault, D.R. and Trejo, L.J., "3-D sound for virtual reality and multimedia", 2000.
- [2] Amengual Garí, S.V., Schissler, C., Mehra, R., Featherly, S. and Robinson, P., March. "Evaluation of real-time sound propagation engines in a virtual reality framework". In Audio Engineering Society Conference: AES International Conference on Immersive and Interactive Audio. Audio Engineering Society, 2019.