주변 잡음이 있는 환경에서 마이크 입력 값을 얻고 가공하는 방법에 대한 연구

김창현[O], 방정원[*]
[O*]청강문화산업대학교 게임콘텐츠스쿨
e-mail: hyun1464@naver.com[O], jwbang@ck.ac.kr[*]

A study on how to process microphone input value in environment with ambient noise

Chang-Hyun Kim[O], Jung-Won Bang[*]
[O*]School of Game, Chungkang College of Cultural Industries

● 요 약 ●

휴대폰에는 기본적으로 마이크가 탑재되어있다. 이것을 어플리케이션에 활용한다면 다양한 콘텐츠를 만들어 낼 수 있고 그러한 콘텐츠가 가진 가능성은 무궁무진 할 것이다. 게임 엔진을 사용하면, 제공되는 마이크 입력 클래스를 사용하여 마이크 입력 값을 가공할 수 있는데, 주변 잡음을 구분하지 하고 소리 입력 값을 받아들이는 문제가 발생한다. 본 논문에서는 기기에 탑재되어 있는 마이크에 접근, 각종 마이크가 출력하는 값을 얻어와 그 값을 가공하는 것을 분석하고, 주변 잡음을 구분하지 않고 받아들이는 문제를 해결하기 위하여 미리 주변 잡음을 체크하고 그 이상의 소리만 입력받도록 하는 방법에 대하여 연구하였다.

키워드: 사운드(sound), 마이크로폰(microphone), 주변잡음(ambient noise)

I. Introduction

휴대폰에는 기본적으로 마이크가 탑재되어있다. 이것을 어플리케이션에 활용한다면 다양한 콘텐츠를 만들어 낼 수 있고 그러한 콘텐츠가 가진 가능성은 무궁무진 할 것이다. 그러한 것들을 실현하기 위하여 Unity의 Microphone 클래스를 사용하여 기기에 탑재되어있는 마이크에 접근, 각종 마이크가 출력하는 값을 얻어와 그 값을 가공하는 것을 구현하였으며, 이렇게 가공한 데이터를 콘텐츠를 위한 수단으로 사용하는데 있어 생기는 문제를 해결하기 위하여 몇 가지의 방법을 구현하였다.

II. Preliminaries

1. Related works

Microphone 클래스의 devices를 사용하여 기기에 연결되어 있는 마이크의 이름을 string 값으로 받아올 수 있다.

```
if (micro_ == null)
    micro_ = Microphone.devices[0];
```

devices는 현재 연결되어 있는 마이크 디바이스의 모든 이름이 저장되어 있는 배열을 반환하는 함수이다.

Microhpone 클래스의 Start 함수를 사용하여 첫 번째 인자로 입력된 마이크가 입력된 사운드 데이터를 녹음하도록 한다. 이렇게 녹음하는 작업을 레코딩을 한다고 한다.

GetPosition 함수를 사용하여 레코딩 샘플 위치가 어디인지 확인한 후 현재 레코딩 중인 마이크의 사운드데이터 샘플을 가지고 있는 Audio Clip의 확인한 GetPosition의 위치부터 샘플을 읽도록 한다.[2]

AudioClip클래스의 GetData는 클립에서 샘플 데이터의 배열을 가져오는 메소드이며, 인자로는 float[] data, int offsetSamples가 있다.[1]

```
for (int i = 0; i < spectrum.Length; i++)
{
    float peak = Mathf.Abs(spectrum[i]);
    if (levelArr[inputLevelCount] < peak)
```

사운드 데이터의 샘플을 얻은 후 사운드 데이터 샘플이 저장된 배열을 순회하며 그 값을 콘텐츠에 맞는 값으로 가공하는 과정을 거친다.먼저 절대 값을 취하여 peak에 저장하고 반복하는 동안 해당 샘플에 가장 큰 진폭을 얻는다.peak을 구하는 데에 있어 절대 값을 취하는 이유는 다음과 같다.

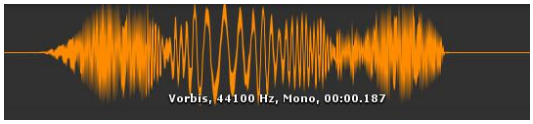

Fig. 1. 사운드 파일

컴퓨터 데이터로 입력된 사운드 파일은 위와같이 위 아래로 진동하며 그 데이터를 저장한다. 진폭이 클수록 큰 소리이며, 이 값은 진동이 위로 갔을 경우 양수로, 아래로 갔을 경우 음수로 저장이 된다. 현재 구현하고 있는 데시벨을 구하는 것을 구현하고 있으며, 데시벨은 음수가 될 수 없기 때문에 음수로 저장되어 있는 데이터는 양수가 되어야 한다. 그러므로 절대 값을 얻기 위하여 Abs연산을 한 것이다.

III. The Proposed Scheme

위의 방식을 그대로 적용하면 데시벨을 체크하는 기능을 구현할 수 있다. 그러나 게임 등의 어플리케이션에서 사용하기 위해서는 고려해야 하는 문제가 있다. 게임에는 배경음, 효과음 등의 추가 사운드가 있으며, 게임을 즐기는 환경이 언제나 조용한 실내가 아니었고 그 문제를 해결하기 위하여 추가적인 처리를 해야 한다.

그 해결 방법으로 두 가지를 생각해 볼 수 있다.

첫째는, 주변 잡음의 데시벨을 체크하여 주변 잡음 이상의 데시벨만을 인식하도록 하는 방법이다. 게임을 시작하고 일정시간 주변 사운드를 체크하여 그 이하 사운드는 주변 잡음으로 체크하고 걸러버리는 방법이다.

```
IEnumerator SetStartNoise()
{
    int loopVal = 10; // 테스트 횟수

    float maxLevel = 0.0f;  // 테스트 중 최대 데시벨

    for (int i = 0; i < loopVal; i++)
    {
        if (level_ > maxLevel) // 최대 데시벨을 넘겼을 시
        {
            maxLevel = level_; // 갱신
        }

        yield return new WaitForSeconds(0.1f);
        // 다음 테스트까지 딜레이
    }

    cutVal = maxLevel + 0.075f; // 주변음 데시벨 설정

}
```

위의 메소드는 주변 잡음을 체크하고 기록하기 위한 메소드이다. 체크할 횟수를 정하고 일정시간 시간을 두고 반복하여 최대 데시벨을 체크한 후 최댓값에 일정 값을 가산하여 cutVal에 그 값을 기록한다.

이후 메인 로직에서 cutVal 이하의 데시벨을 0으로 만들어주면 주변 잡음이 인식되는 문제를 해결할 수 있다.

둘째로, 외부 마이크를 사용하는 방법이다. 이 방법은 주변 잡음이 마이크가 반환하는 값의 최댓값에 근사한 값까지 올라갔을 경우 사용해야 하는 특수한 케이스를 위한 방법이다. 공연, 행사장, 게임 전시회 등의 주변 잡음이 비정상적으로 큰 경우 첫번째의 방법을 사용한다고 하여도 언제나 정상적으로 작동할 것이라고 기대하기 어렵다. 그렇기에 그러한 특수한 케이스에는 외부 마이크를 연결하여 외부 마이크의 사운드를 인식시키는 것으로, 마이크가 가지고 있는 노이즈 제거 기능을 사용하면 첫번째 방법과 같이 사용한다고 하였을 때 상당한 안정성을 확인하였다.

IV. Conclusions

마이크는 점점 진화하고 있다. 점점 민감하게 작은 소리도 잡아낼 수 있도록 발전하고 있으며 더 선명하게 그것을 데이터화 시킬 수 있도록 진화하고 있다. 그런만큼 마이크를 이용한 콘텐츠를 제작하는 경우, 주변 소음을 차단하는 방법에 대한 연구가 병행되어야 한다.

REFERENCES

[1] docs.unity3d.com/Manual/index/html
[2] www.yotube.com/watch?v=U7jw2fPkoJU