
사운드 실감성 증진을 위한 사운드 감지 알고리즘 기반 촉각진동자극 생성

Vibration Stimulus Generation using Sound Detection Algorithm for Improved Sound Experience

지동주, Dongju Ji*, 오성진, Sungjin Oh**, 전경구, Kyungkoo Jun**, 성미영, Meeyoung Sung**

요약 적절한 촉각자극은 사운드 효과의 실감성을 증진시킬 수 있다. 예를 들어, 게임이나 영화 등에서 총소리와 함께 발생하는 진동자극은 감상자의 몰입감을 향상시킬 수 있다. 이러한 맥락에서 음향만을 담고 있는 기존 사운드 파일에 촉각자극정보를 추가하여 사운드 플레이와 함께 햅틱장치를 통해 진동자극을 준다면 좀 더 사실감 있는 사운드 체험을 할 수 있을 것이다. 본 논문에서는 사운드 파일을 분석하여 그에 따른 촉각자극정보를 생성하는 방법을 제안한다. 촉각자극정보란 사운드 파일 내에서 특정 소리가 발생하는 시간위치를 발생시켜야 하는 촉각자극패턴이다. 수작업으로 이러한 촉각자극정보를 생성하는 것은 많은 수고를 필요로 한다. 본 논문은 이러한 수작업을 대체할 수 있도록, 사운드 파일에서 특정 소리의 발생시점을 검색할 수 있는 사운드 감지 알고리즘과 촉각자극정보 생성방식을 제안한다. 사운드 감지 알고리즘은 검색하고자 하는 사운드의 주파수 특성을 이용하여 유사한 특성을 가지는 시점들을 찾아낸다. 개발한 알고리즘을 실험한 결과, 5가지 서로 다른 사운드에 대해 98% 이상의 감지 성공률을 얻을 수 있었다. 연구 결과를 종합하여 GUI기반의 진동패턴 생성 유틸리티를 개발하였다. 이 유틸리티를 이용하면 사운드 파일에서 특정 소리의 발생 부분을 자동으로 검색하고, 그 시점마다 원하는 진동정보를 손쉽게 생성할 수 있다.

Abstract Sound effects coming with appropriate tactile stimuli can strengthen its reality. For example, gunfire in games and movies, if it is accompanied by vibrating effects, can enhance the impressiveness. On a similar principle, adding the vibration information to existing sound data file and playing sound while generating vibration effects through haptic interfaces can augment the sound experience. In this paper, we propose a method to generate vibration information by analyzing the sound. The vibration information consists of vibration patterns and the timing within a sound file. Adding the vibration information is labor-intensive if it is done manually. We propose a sound detection algorithm to search the moments when specific sounds occur in a sound file and a method to create vibration effects at those moments. The sound detection algorithm compares the frequency characteristic of specific sounds and finds the moments which have similar frequency characteristic within a sound file. The detection ratio of the algorithm was 98% for five different kinds of gunfire. We also develop a GUI based vibrating pattern editor to easily perform the sound search and vibration generation.

핵심어: *Sound detection, Haptic, FFT, Vibration, Signal processing, Tactile*

본 논문은 2008년 인천정보산업진흥원의 지역산업기술혁신사업 실감형 3D 영상 원천기술 개발 과제로부터 지원받았음.

*주저자 : 인천대학교 멀티미디어시스템공학과 석사과정 e-mail: chidj11@incheon.ac.kr

**공동저자 : 인천대학교 멀티미디어시스템공학과 석사과정 e-mail: oro777@incheon.ac.kr

**공동저자 : 인천대학교 멀티미디어시스템공학과 교수 e-mail: kjun@incheon.ac.kr

**공동저자 : 인천대학교 컴퓨터공학과 교수 e-mail: mysung@incheon.ac.kr

1. 서론

사운드와 함께 적절한 촉각자극을 제공하면 실감성을 증진시킬 수 있다. 예를 들어, 게임이나 영화 등에서 총소리와 같은 효과음과 함께 진동자극이 발생하면 감상자는 해당 상황에 더욱 몰입할 수 있다[1]. 이에 따라 사운드에 반응하여 진동을 발생시키는 인터페이스 장치들이 많이 개발되고 있는데, 진동헤드셋[2]이 대표적인 예이다.

이러한 기존 진동 인터페이스는 특정 주파수 영역에 해당하는 사운드가 발생할 때 한정된 패턴으로만 진동을 발생시키기 때문에, 사용 초기에 향상되었던 실감효과가 급속히 쇠퇴하는 단점이 있다.

본 논문은 영화나 게임의 사운드 파일을 분석하여 그에 따른 촉각자극정보를 손쉽게 생성하는 방법을 제안한다. 촉각자극정보란 사운드 파일 내에서 특정 소리가 발생하는 시점과 발생시켜야 하는 촉각자극패턴으로 구성된다. 본 논문에서는 촉각자극을 위해 진동을 이용한다.

진동 인터페이스가 부착된 PC에서 이러한 촉각자극정보를 이용하면 사운드 파일을 듣는 동시에 촉각효과도 제공할 수 있어, 실감성 향상에 기여할 수 있다.

사운드 파일에 대한 촉각자극정보를 일일이 수작업으로 생성하려면 많은 시간과 수고가 필요하다. 이를 해결하기 위해 본 논문에서는 사운드 파일에서 특정 소리의 발생시점을 검색할 수 있는 사운드 감지 알고리즘과 촉각자극정보 생성 방식을 제안한다.

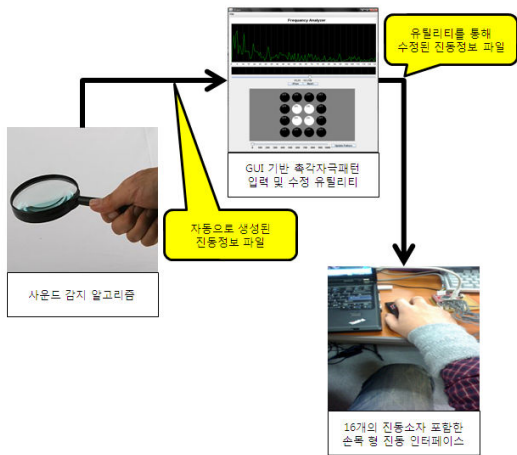


그림 1 시스템의 전체적인 그림

사운드 감지 알고리즘은 주파수 특성을 이용한다. 특정 사운드의 주파수 분포 특성을 미리 분석하고, 사운드 파일에서 유사한 특성을 가지는 시점들을 찾아낸다. 이러한 알고리즘은 진동패턴생성 유틸리티에 적용되어, 사운드 파일에서 특정 사운드 발생시점을 검색하고, 그 시점들마다 적절한 촉각자극정보를 생성하는 데 사용된다. 또한 GUI 기반의 이

유틸리티는 수작업으로 촉각자극패턴을 입력 또는 수정하는 과정을 용이하게 해준다. 사운드 파일 내에서 특정 사운드가 발생하는 시점을 자동으로 찾아서 촉각자극패턴을 생성하고, 이 유틸리티를 이용해 잘못된 패턴에 대해서 입력 및 수정을 한다.

제안하는 사운드 감지 알고리즘의 정확성을 검증하기 위해 First Person Shooting(FPS) 게임의 5가지 총소리를 구분하고 각각에 맞는 진동효과를 주는 실험을 수행하였다.

2. 사운드 감지 알고리즘

사운드 감지 알고리즘은 사운드 파일 내에서 발생하는 특정사운드의 발생 시점을 찾아내고, 그 시점에 해당하는 촉각자극정보를 생성하는 알고리즘이다.

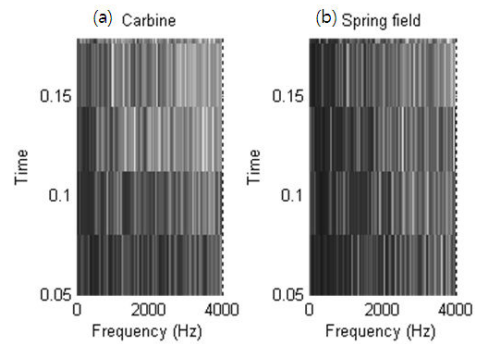


그림 2 시간 단위 주파수 분포의 차이

제안 알고리즘의 핵심은 감지하고자 하는 소리의 주파수 분포 특성을 미리 분석하고, 그를 토대로 실제 사운드 파일 내에서 비슷한 주파수 특성을 갖는 부분을 찾아내는 것이다. 주파수 분포 특성을 비교할 때 시간 별로 주파수분포가 다르게 나타나는 점을 이용한다. 예를 들어, 그림 1은 두 가지 다른 총소리에 대한 시간 별 주파수특성을 스펙트럼으로 표현한 것이다. Y축은 시간의 흐름을 나타내고, X축은 주파수 영역을 나타낸다. 색이 진할수록 해당 주파수영역의 특성이 강하다는 것을 뜻한다. (a) 스펙트럼은 저주파에서의 특성이 시간이 지남에 따라 많이 감소된다. 그에 반해, (b) 스펙트럼은 저주파에서의 특성이 비교적 많이 유지되는 것을 볼 수 있다.

사운드가 가지는 주파수 분포 특성을 알아내기 위해 타임도메인의 신호를 주파수도메인의 신호로 바꿀 수 있는 Fast Fourier Transform(FFT)을 사용한다[3]. FFT를 수행하면 각 주파수 영역에 대한 특성 값인 magnitude 들이 나온다.

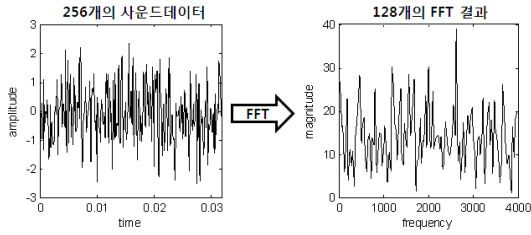


그림 3 FFT 수행의 예

예를 들어, 그림 2처럼 8 KHz로 샘플링된 사운드에서 256개의 샘플데이터를 모아 FFT를 수행하면, 0~4000Hz 주파수범위를 128개의 동일 크기 영역으로 나누고, 그 각 영역에 대한 magnitude가 계산된다. 이 때, 한 번의 FFT가 수행되는 256개의 샘플데이터 모음을 윈도우라고 한다.

이러한 FFT를 이용해서 사운드의 주파수 분포 특성을 분석하다보면, 동일한 사운드일지라도 조금씩 분포 특성이 달라지는 경우가 생긴다. 예를 들어, 게임의 같은 총소리일지라도, 반복 때마다 주파수 분포가 조금씩 달라진다. 정확한 원인은 알 수 없었지만, 게임에서 사용하는 사운드 발생 방식에 기인한 것으로 추측된다. 이 경우, 동일한 총소리일지라도 특징이 나타나는 주파수 영역이 좌우로 분산되는 현상이 발생한다.

사운드 감지 알고리즘에서는 이러한 현상을 고려하여, 사운드의 주파수 특성을 추출할 때, 특징이 분산된 좌우 주파수 영역의 magnitude도 분석에 이용한다. 예를 들어, 특징이 나타나는 주파수 영역이 F_{n-1} 이고, 해당 magnitude가 X_{n-1} 이라고 할 때, 분석에 사용하는 magnitude 값은 수식 (1)과 같이 연산한다.

$$Y_{n-1} = X_{n-2} + X_{n-1} + X_n \quad (1)$$

사운드의 볼륨은 FFT 연산 결과를 비례적으로 왜곡시킨다. 즉, 볼륨이 클수록 magnitude값도 커진다. 이를 해결하기 위해 normalization 과정을 수행한다. 이 과정에서는 FFT 결과로 나온 magnitude값들을 가장 큰 magnitude값으로 나눈다.

또한, 사운드 감지 알고리즘의 성능향상을 위해 사운드 검색과정의 연산량을 줄인다. 특정 사운드 비교를 위해 모든 주파수 영역의 magnitude에 대해 비교를 수행하지 않고, 제안한 알고리즘에서는 특성을 표현할 수 있는 n개의 주파수 영역에서의 magnitude만을 비교한다. 현재 구현된 알고리즘에서 $n = 4$ 이다.

제안 알고리즘은 두 부분으로 구성되어 있다. 감지하고자 하는 사운드의 주파수 분포 특성을 분석하는 부분과 분석한 정보를 토대로 사운드데이터에서 특정사운드의 발생 시점을 감지하는 부분이다. 두 부분에 대한 자세한 설명은 2.1장과

2.2장에서 설명한다.

2.1 주파수 분석

알고리즘의 첫 번째 부분인 사운드의 주파수 분포 특성 분석 과정은 그림 3과 같이 수행된다. 여기서는 총소리 분석의 예를 사용하여 설명한다. 총소리 발생 시작지점부터 1024개의 데이터를 모은다. 그 데이터를 256개씩 4개의 윈도우로 나누어 각각 FFT를 수행한다. 그리고 주변의 값을 합치고 normalization을 수행한다. 시간의 흐름에 따른 윈도우 4개의 FFT 결과가 생성된다. 규칙적으로 나타나는 magnitude를 찾기 위해 총소리를 여러 번 녹음해서 같은 작업을 수행한다. 결과로 나오는 magnitude 중 가장 큰 값, 즉 1이 나오는 곳이 peak가 된다. 여러 번 수행한 결과에 대해 같은 인덱스의 magnitude에서 가장 작은 값과 가장 큰 값의 차이가 적을수록 규칙적으로 나타난다고 판단한다. magnitude가 가장 규칙적으로 나오는 곳의 인덱스와 그 값을 비교대상으로 선택한다. 결과적으로 4개의 윈도우에 대해 각각 peak와 4개의 비교 값이 나온다.

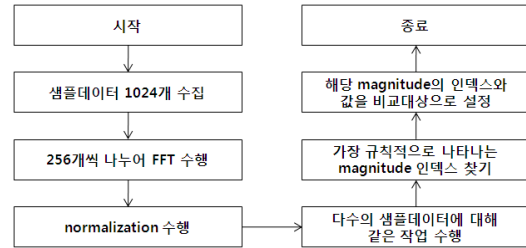


그림 4 주파수 분석 순서도

2.2 특정 사운드 검색

알고리즘의 두 번째 부분인 특정 사운드 검색 과정은 그림 4와 같은 순서로 이뤄진다. 이 과정은 1024크기의 윈도우를 슬라이딩하는 방식으로 이루어진다. 우선 1024개의 데이터를 모아서 4개의 윈도우로 나눠 각각 FFT를 수행한다. 주변의 값을 합치고 normalization을 수행한다. 4개의 윈도우에 대한 FFT 결과가 나오면 각각에서 가장 큰 magnitude가 나오는 인덱스를 찾아낸다. 4개의 peak 모두 미리 분석한 peak와 같다면 다음 단계로 넘어간다. 그리고 비교 값으로 설정한 인덱스의 magnitude를 뽑아낸다. 분석을 통해 미리 저장해놓았던 magnitude와 사운드데이터를 분석해서 얻은 magnitude 간의 차이를 계산한다. 그 차이의 절대값을 계산하고 각 윈도우 별로 그 절대값을 모두 합한다. 각 윈도우의 계산 결과가 미리 설정해 놓은 threshold보다 작다면 감지하려는 소리로 판단하는 것이다. 이 알고리즘의 마지막 단계에서는 특정 사운드 발생이 감지되었을 경우, 진동정보를 생성한다. 진동정보 포맷에 대해서는 다음 절에서 자세히 설명한다.

다. 위의 과정이 종료되면, 윈도우를 16개 샘플데이터 만큼 슬라이딩시키고, 다시 동일 과정을 반복한다.

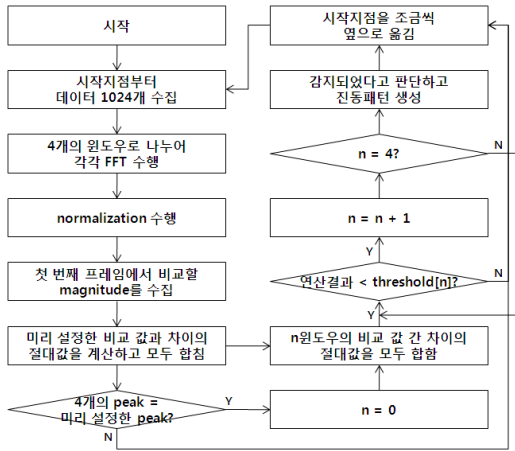


그림 5 특정 사운드 검색 순서도

2.3. 실험결과

게임에서 발생하는 총소리 5개를 녹음을 하여 자동감지 알고리즘의 정확성을 검증했다. 검증에 사용한 사운드데이터는 그림 5와 같다.

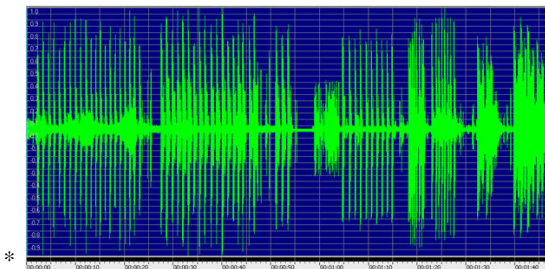


그림 6 실험에 사용한 사운드데이터

검증 결과 그림 6과 같은 결과를 도출했다. 모든 소리에 대한 감지율은 66번의 소리 중 1번만 감지하지 못해 98% 이상인 것을 확인 할 수 있었다.

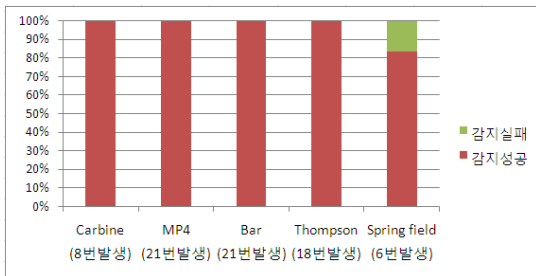


그림 7 검증 결과 얻은 감지율

3. 진동패턴 생성기

연구 결과를 종합하여 GUI기반의 진동패턴 생성 유틸리

티를 개발하였다. 이 유틸리티를 이용하면 사운드 파일에서 특정 소리의 발생 부분을 자동으로 검색하고, 그 시점마다 원하는 촉각자극정보를 손쉽게 생성할 수 있다.

이 유틸리티는 WAV 파일을 로드해서 사운드 감지 알고리즘을 적용한다. 그 결과 생성되는 진동정보는 다음과 같은 포맷을 가진다.



그림 8 진동정보파일의 포맷

진동정보파일은 다수의 촉각자극정보로 구성된다. 촉각자극정보는 그림7과 같이 해당 프레임의 사이즈와 진동이 울려야하는 시간, 진동패턴과 지속시간을 포함한다. 진동패턴은 진동인터페이스에 대한 진동을 올리는 패턴에 대한 정보를 가지고 있다. 이 때, 한 프레임 내에 진동패턴과 지속시간은 다수가 될 수 있다. 프레임의 사이즈는 자기 자신을 포함한다. 사운드가 재생되고 있을 때, 해당시간이 되었을 경우 패턴과 지속시간대로 진동을 발생시킨다.

사운드 감지 알고리즘을 통해 나오는 진동정보파일에는 잘못된 정보가 들어있을 수 있다. 감지하지 못했을 경우와 잘못 감지한 경우가 있다. 이 경우 진동패턴 생성기 유틸리티의 GUI를 이용하여 제거 및 추가를 손쉽게 할 수 있다.

진동패턴 생성기는 다음과 같은 순서로 수행한다. 우선 WAV 파일을 읽어서 사운드 감지 알고리즘 프로그램과 연결시켜 진동정보파일을 미리 생성한다. 화면을 통해 진동의 발생시점과 해당 진동패턴을 알 수 있다. 그 중에서 감지하지 못한 부분에 대해서 직접 진동패턴을 입력하고, 잘못 감지한 부분에 대해서 진동패턴을 지운다.

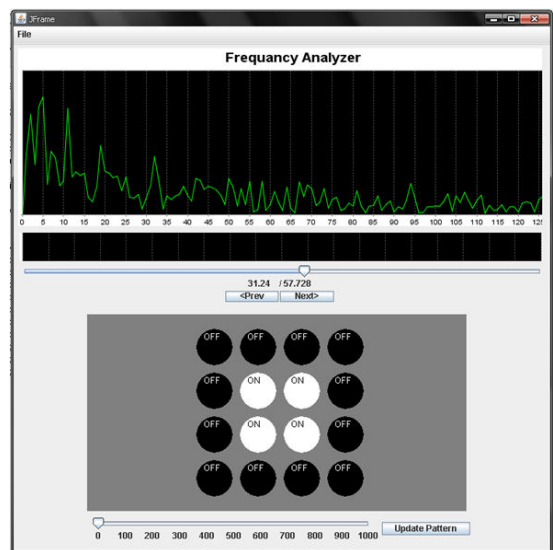


그림 9

GUI는 그림8과 같이 구성된다. 현재 위치부터 데이터를 모아 256-FFT를 수행한 결과가 나오게 된다. 또한 진동패턴이 발생하는 시점을 알 수 있다. 시간 단위로 이동을 하면서 적절한 시점에 진동인터페이스의 진동 패턴과 지속시간을 설정해서 패턴을 업데이트할 수 있다.

4. 결론

본 논문에서는 사운드데이터에 대해서 적절한 진동정보파일을 생성할 수 있는 프로그램을 구현했다. 진동패턴 생성기를 통해서 파일을 생성할 수 있고, 사용자가 진동패턴을 일일이 입력하는 불편함을 없애기 위해 사운드 자동감지 알고리즘을 적용했다. 이 알고리즘을 사용해서 98% 이상의 감지율을 얻어냈다. 감지하지 못하거나 잘못 감지한 경우에 대해서는 GUI 기반 촉각자극정보 입력 및 수정 유틸리티를 통해 진동정보를 수정할 수 있었다.

사운드 자동감지 알고리즘을 사용한 진동패턴 생성기는 사용자가 사운드를 들으면서 적절한 진동효과도 제공을 해 현실감을 상승시킬 수 있도록 도울 수 있다.

참고문헌

- [1] Fuller C.R. and von Flotow A.H., "Active control of sound and vibration" Control Systems Magazine, IEEE, pp. 9~19, 1995
- [2] <http://www.hyundai-jpc.com/>
- [3] Soontorn Oraintara, Yung-Jui Chen and Q. Nguyen, "Integer Fast Fourier Transform" IEEE Transactions on Signal Processing, Vol. 50, No. 3, 2002