EM 최적화를 이용한 오디오 텍스처 합성

Audio Texture Synthesis using EM Optimization

노창환, Chang-Hwan Roe*, 유민준, Min-Joon Yoo**, 이인권, In-Kwon Lee***
연세대학교 컴퓨터과학과

요약 오디오 텍스처 합성은 주어진 짧은 오디오 클립으로부터 임의의 길이를 갖는 새로운 오디오 클립을 생성하는 방법이다. 이는 애니메이션이나 영화에서 비디오와 정확한 동기화를 이루는 사운드 효과를, 혹은 임의의 길이를 갖는 배경 음악을 효율적으로 만들 수 있는 방법이다.

최근 Lie Lu 는 주어진 예제 오디오 클립을 여러 조각으로 나눈 후, 이 조각들을 그래프 형태로 연결하고, 생성된 그래프를 탐색하면서 임의의 길이를 가지는 오디오 클립을 합성하는 방법을 제안하였다. 비교적 간단한 방법으로도 원본 오디오 클립과 비슷한 느낌의 오디오 클립을 만들어낸다는 장점이 있지만, 이는 원본 내의 여러 오디오 조각들이 단지 지속적으로 연결되는 형태로 합성되기 때문에 종종 반복되는 느낌을 받는다는 단점이 있다.

본 논문에서는 Lie Lu 의 방법과는 달리 주어진 예제 오디오 클립을 직접 합성함으로써 반복성을 줄이면서도 원본과 비슷한 느낌을 갖는 결과 오디오 클립을 생성할 수 있는 방법을 제안한다. 특히 본 논문에서는 정확한 합성을 위하여 EM 최적화 방법을 사용한다. 본 논문에서 제안하는 합성 방법은 먼저 예제 오디오 클립을 일정 단위로 나누고 이렇게 나눠진 부분들을 일정 길이만큼 서로 겹쳐지게 합성하여 임의의 길이의 오디오 클립을 만든다. 그 후 만들어진 오디오 클립을 예제 오디오 클립과 부분 부분을 비교하여 확장된 오디오 클립과 최대한 비슷한 부분을 예제 오디오 클립에서 찾는다. 그 다음 찾아진 결과를 결과 오디오에 다시 합성하여 오디오 클립을 만든다. 이런 과정을 반복하여 최적화된 가장 적절한 결과값을 구한다. 이 결과는 분할된 부분들이 가장 자연스럽게 이어지는 결과가 된다.

본 논문에서는 최적화를 사용하여 오디오를 합성하기 때문에 합성 결과를 쉽게 조정할 수 있다는 장점이 있다. 최적화 문제에 특정 제약 조건을 넣음으로써 사용자가 원하는 부분의 음악이 결과 사운드의 특정 부분에 위치 할 수 있게 하고 이로써 특정 흐름을 만들어낼 수 있으며, 일부가 손실된 사운드 데이터의 복구를 가능하게 하는 등의 결과를 생성할 수 있다.

EM 최적화를 사용한 오디오 텍스처 합성 방법은 기존의 합성 방법에 비해 질적인 측면에서 보다 좋은 결과를 생성할 수 있고, 비교적 반복이 덜한 패턴들을 만들어 낼 수 있다. 이를 입증하기 위해 이에 대한 사용자 설문 조사 결과가 제시된다.

핵심어: : Audio Texture Synthesis, Sound Synthesis EM Optimization

본 연구는 문화관광부 및 한국문화콘텐츠진흥원의 문화콘텐츠기술연구소(CT) 육성사업의 연구결과로 수행되었음.

1. 서론

오디오 미디어는 게임, 애니메이션 혹은 영화의 배경음악과 특수음향을 비롯한 거의 대부분의 멀티미디어 매체에서 중요한 역할을 하는 요소 중에 하나이다. 주로 비디오와 같은 시각적인 효과에 맞물려져 재생되거나 혹은 멀티미디어 매체 자체 내에서 중심적인 역할을 담당한다. 예를 들어 게임의 경우 특수효과와 같은 시각적인 요소도 중요하지만 배경 음악이 덧붙여진다면 이를 이용하여 주인공이 위치하고 있는 장소에 대한 정보를 전달하여 주거나 게임의 분위기를 더욱 강조하는 등 사용자가 게임에 더욱 몰두하게끔 할 수 있다.

이런 멀티미디어 매체 내 요소들의 중요한 결합 인자는 각각 요소들의 시간의 동기화이다. 즉 각각의 멀티미디어들이 적절히 동기화된 길이를 갖고 서로 연관된 시간에 재생이 되어야 사용자의 몰입도가 증가하게 된다.

비디오를 비롯한 다른 멀티미디어 요소에 비해 오디오가 가지는 본질적인 단점은 원본 오디오의 길이를 자유롭게 조정하는 것이 힘들다는 점이다. 비디오는 간단한 보간법으로 길이를 조정할 수 있지만, 오디오의 경우는 음정 자체가 진동(frequency)으로 되어있기 때문에 시간의 변화는 필연적으로 오디오의 음정의 변화를 가지고 오게 된다.

음정의 변화 없이 오디오의 길이를 변화시키는 방법 중 한가지는 PSOLA[1]가 있다. 이는 오디오를 아주 작은 부분들로 나누어 이를 복사하고 붙임으로써 음정의 변화 없이 오디오의 길이를 변화시키는 방법이다. 하지만 이 방법으로는 어느 한도 내의 짧은 길이의 합성만이 가능하다.

다른 맥락을 가진 방법이지만 A. Zils[2]는 방대한 데이터베이스에서 사운드 예제를 가져와 조합하여 새로운 오디오 데이터를 만들어 내는 방법을 제안하였다.

오디오의 길이를 변화시키는 방법 중 또 한가지는 Lie Lu가 제안한 'Audio Textures'[3]가 있다. 이 방법은 A. Schödl의 'Video Textures'[4] 논문에서 비디오의 시퀀스가 자기 유사성을 가지고 있기 때문에 이 시퀀스를 적절히 여러 조각으로 잘라낸 후 조각들 사이를 그래프 형태로 연결한 후, 이 그래프를 탐색하여 임의의 길이를 가진 비디오를 만들 수 있다는 점을 착안하였다. 즉 Lie Lu는 오디오 역시 자기 유사성을 가지고 있기 때문에 원본 오디오를 적절히 잘라내어 만들어진 그래프들 간의 탐색을 통하여 임의의 길이를 가진 오디오를 만들 수 있다고 가정하였다. 이 방법은 짧은 원본 오디오에서도 다양한 결과를 만들어 낼 수 있고 더불어 만들 수 있는 결과 오디오의 길이에 대한 제한이 없지만, 역시 기존의 방법인 복사와 붙이기 방법에서 크게 벗어나지 못하기 때문에 종종 반복적인 느낌을 받는다. Lie Lu는 음악의 자기 유사성을 측정하기 위하여 Foote의 노벨티 스코어(Novelty Score) [5]를 이용하였다.

본 논문이 제안하는 오디오 텍스처 합성법은 Vivek Kwatra 의 'Texture Optimization for Example-based Synthesis'[6] 논문에서 아이디어를 얻었다. 이 논문에서는 G. McLachlan [7]이 제안한 EM 최적화 방법과 유사한 EM 최적화 방법을 이용하여 예제 이미지 텍스처를 합성하여 임의의 크기를 갖는 결과 이미지 텍스처를 생성하는 방법을 제안되었다. 우리는 이미지 텍스처의 합성이 결국 원본 이미지 텍스처에 포함되어있던 신호들을 가져와 적절히 나열하여 만들어지는 것이기 때문에 역시 오디오 텍스처도 원본 오디오에 포함되어있는 신호들을 가져와 이미지 합성 방법과 비슷한 방법으로 신호들을 나열하여 결과를 얻을 수 있다고 가정하였다.

본 논문에서는 기존의 방법과는 달리 최적화 방법을 이용하여 오디오를 합성함으로써 반복성을 줄이면서도 원하는 길이를 갖는 오디오를 생성하는 방법을 제안한다. 특히 이 논문에서는 EM 최적화[6, 7]를 사용하여 정확한 합성이 이루어지도록 하였다. EM 최적화는 최적화에서 사용되는 데이터와 최적화의 변수를 모두 알지 못할 때, 두 가지 단계의 최적화를 연속적으로 행함으로써 최종 최적해를 찾는 방법이다. 최적화를 사용한 합성의 가장 큰 특징 중 하나는 합성 시 제약 조건을 주어 결과를 조정할 수 있다는 것이다. 따라서 우리의 방법은 기존 방법보다 더욱 간단하면서도 효과적으로 사용자가 원하는 오디오 결과를 유도 및 생성할 수 있게 된다.

2. 시스템 개요

EM 최적화를 사용한 오디오 텍스처 합성은 크게 두 가지 과정으로 요약될 수 있다. 즉 오디오 텍스처의 에너지를 최소화하는 과정과 예제 오디오 텍스처와 생성된 오디오 텍스처의 가장 가까운 이웃들의 집합을 구하는 과정 두 가지 이다. 실질적으로 결과는 이 두 과정을 반복함으로써 얻어지게 된다.

전체적인 시스템의 구조는 다음 그림1과 같다.

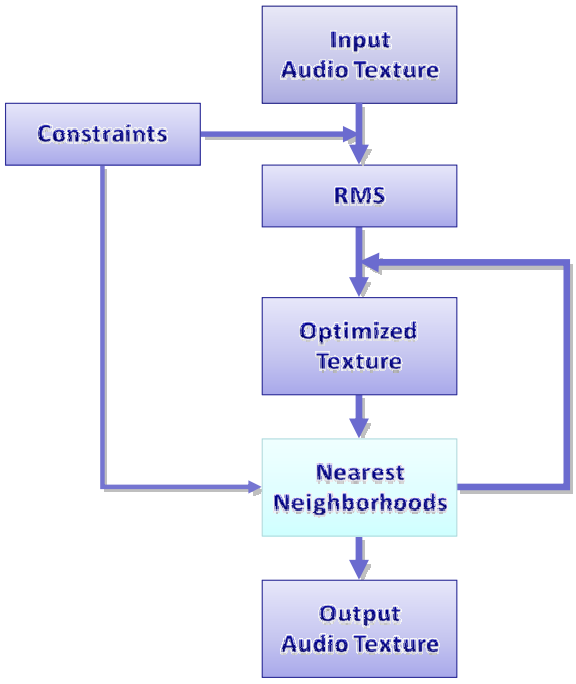


그림 1 시스템 개요

3. 오디오 텍스처 평균화

오디오 시그널은 매우 짧은 시간을 단위를 기준으로 하여 단위 시간마다 진폭 값이 기록되는 형태로 되어있다. 진폭 값이 기록되는 단위 시간의 간격은 굉장히 짧기 때문에 이렇게 얻어지는 데이터를 그대로 사용하기 위해서는 많은 계산 비용이 필요하게 된다. 따라서 계산 전에 일정 시간 혹은 일정 개수 단위로 평균화하여 데이터들의 수를 줄인다. 본 논문에서는 EM 합성 계산시에 오디오 시그널에 0.1초의 시간을 간격으로 RMS(Root Mean Square)를 적용한 값을 사용한다. z_p 는 예제 오디오 텍스처의 진폭 값이며 x_p 는 결과 오디오 텍스처의 진폭 값일 때, 계산시에는 이의 RMS 버전인 $z_p^{'}$, $x_p^{'}$ 를 사용한다.

$$z_{ip}^{'} = \sqrt{\frac{1}{N}\sum_{i=0}^{N} z_p}\ , \qquad x_{ip}^{'} = \sqrt{\frac{1}{N}\sum_{i=0}^{N} x_p} \tag{1}$$

실제 최종적으로 얻어지는 결과 오디오는 평균화되지 않는 원본 데이터를 직접 합성한다.

4. 오디오 텍스처의 유사성과 합성

본 논문에서는 예제 오디오 텍스처와, 최적화 과정 중간에 합성된 결과 오디오 텍스처와의 유사성을 수치화함으로써 오디오 텍스처의 에너지를 수치적으로 가늠하게 하였다. 전체 에너지는 예제 오디오 텍스처의 지역 이웃들과 결과 오디오 텍스처의 지역 이웃들과의 유사성의 합으로 정의한다. 많은 픽셀기반 이미지 합성 혹은 패치 기반 이미지 합성 방법[8, 9]들은 이미지는 원본 이미지의 형태를 유지하기 하여 각각의 픽셀을 원본에서 유사한 이웃들과 비교하여 가장 잘 어울리는 픽셀을 구하는 방법을 사용한다. 오디오도 이와 유사한 방법을 사용하여 구할 수 있다. 하지만 우리는 이런 지역적인 비교 및 합성 방법도 고려하면서도 이를 통해 전체 에너지를 한 번에 구할 수 있는 방법을 고안하였다.

이미지에서는 각 픽셀 정보 하나하나가 색이라는 정보로 인식되는 반면, 사운드에서는 어느 정도 수 이상의 샘플링 값이 소리 정보로 인식되게 된다. 따라서 본 논문에서는 어느 정도로 인지될 수 있는 음을 들려줄 수 있을 만큼의 시간 단위를 사용한다. 하나의 이웃에 대한 에너지는 결과 오디오 텍스처의 이웃과 예제 오디오 텍스처의 이웃과의 거리로 정의한다. 결국 전체 에너지는 이런 지역적 이웃들의 에너지의 합이라고 볼 수 있다.

X 는 사용자가 원하는 결과 즉 합성되는 오디오 텍스처를 의미하고, Z 는 예제 오디오 텍스처를 의미한다고 하자. 그리고 x 와 z 는 각각 X 와 Z 를 벡터화 한 것이다. 즉, X 와 Z 의 전체 샘플의 진폭 값을 의미한다. w 는 이웃의 너비를 의미하고 N_p 는 p 를 중심으로 주변 진폭 값들을 포함하는 이웃을 나타낸다. N_p 에 대응하는 x 의 부분벡터들은 x_p 로 나타낸다. z_p 는 마찬가지로 z 의 N_p 에 대응하는 z 의 부분벡터들을 나타낸다. 이렇게 정의한 후 결과 오디오 텍스처 X 로부터 얻어지는 에너지를 정의하면 다음과 같다.

$$E_t(x;\{z_p\}) = \sum_{p\in X^{\dagger}} \left\| x_p - z_p \right\|^2 \tag{2}$$

(z_p 와 x_p 는 각각 원본 오디오와 결과 오디오의 샘플들. 보통 6개의 단위(0.6초)의 길이를 가짐.)

$X^{\dagger}$ 는 진폭 값들의 일부 위치를 의미하는데 이는 모든 진폭 값들의 위치를 기준으로 구하는 것이 아니라 일정 간격으로 이웃의 일부만이 겹치게 하여 최적화된 값을 구하는 것을 의미한다. 이를 통하여 중복성을 줄이고 계산시간을 줄이게 된다.

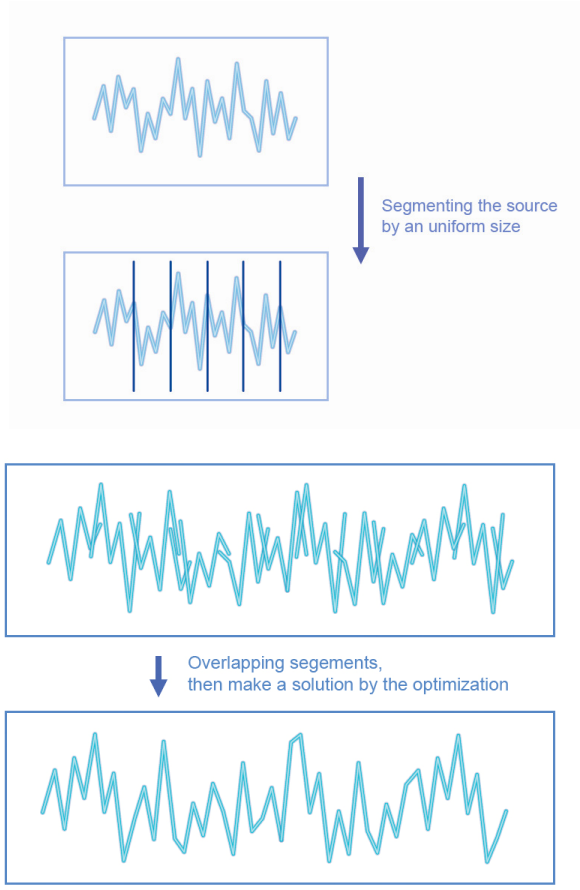


그림 2 최적화 과정

EM 최적화 방법은 크게 두 단계로 나눌 수 있다. 우선 E 단계에서는 결과 오디오 텍스처 X 를 최소화 한다. 이것은 선형 방정식을 이용하여 x 를 0에 가깝게 만듦으로써 문제를 풀 수 있다. 이것은 z_p 와 가장 가까운 x_p 를 구하는 것을 의미한다. 즉, 에너지를 최소화 하는 과정이다.

두 번째 M 단계에서는 x_p 와 가까운 이웃들에 대응되는 리스트를 예제 오디오 텍스처 z_p 에서 구하게 된다. 이 단계는 결과 오디오 텍스처의 이웃들과 가장 가까운 이웃을 예제 오디오 텍스처에서 찾는 문제라고 볼 수 있다. E 단계에서 최적화된 X 에 따라 대응되는 z_p 는 조금씩 변하게 된다.

가장 가까운 이웃들에 대응되는 리스트라고 할 수 있는 z_p 를 구하게 되면 다시 E 단계를 거쳐 z_p 에 대응되는 최

적화된 X 를 생성하고 또다시 X 는 바뀌게 됨으로 z_p 는 계속 새로운 값을 가지게 된다. 이렇게 두 과정(E, M)을 반복하다 보면 z_p 는 어느 시점에서 변하지 않고 고정된 값을 유지함으로써 수렴하게 된다. 우리의 실험으로는 최종 수렴이 이루어지지 않더라도 몇 번의 반복만 행하면 만족할만한 결과를 생성할 수 있었다. 이를 정의한 알고리즘은 다음과 같다.

Audio Texture Synthesis Algorithm

$z_p^0 \leftarrow random\ neighborhood\ in\ Z\ \forall p \in X$

for iteration $n = 0 : N$ *do*

$x^{n+1} \leftarrow \arg\min_x E\ (x, \{z_p^n\})$

$z_p^{n+1} \leftarrow nearest\ neighborhood\ of\ x_p^{n+1}\ in\ Z\ \forall p \in X$

if $z_p^{n+1} = z_p^n\ \forall p \in X$ *then*

$x \leftarrow x_p^{n+1}$

break

end if

end for

4.1 Gaussian fall-off function

추가로 Gaussian fall-off function을 통해서 얻어진 확률을 각각의 진폭 값에 대한 가중치로 적용하였다. 이렇게 가중치를 적용한 이유는 이웃의 중심으로 갈수록 보다 데이터가 결정되는데 중요한 역할을 하기 때문이다. 이 방법은 보통 두 개의 오디오 클립을 연결 시킬 때 쓰이는 교차 페이드(Cross Fade) 효과를 적용한 결과와 유사한 결과를 기대할 수 있다. 또한 이를 응용하여 전혀 다른 이웃들 간의 결합도 가능하게 할 수 있다.

4.2 다단계 합성

우리는 제시한 알고리즘에 다단계 합성 방법[10]을 적용하였다. 먼저 예제 오디오 클립을 낮은 샘플링 비율에서 합성을 한 후 이 결과를 보간을 통해 업샘플링하여 샘플링 비율을 높인다. 이후 업샘플링된 결과를 가지고 다시 알고리즘을 적용하여 하는 식으로 계속 이 과정을 반복한다. 또한 샘플링 비율 증가하는데 반비례하여 이웃의 크기를 축소함으로써 이웃 내의 데이터의 개수를 조절해 나간다.

결과론적으로 보면 EM 최적화 방법은 초기값들에 민감하다. 그 이유는 결과물이 초기값을 가진 상태에서 시작하여 가장 초기값과 가까운 값들을 지닌 이웃들을 원본에서 찾고 원본에서 찾은 부분을 바탕으로 에너지의 합이 최소가 되는 결과물을 만들어낸다. 이때 다시 결과물과 가장 가까운 이웃 값들을 원본에서 찾는데 이전에 찾았던 이웃들과 차이가 없다면 합성은 수렴되고, 달라졌다면 다시 계속 반복하여 계속 이전과 가까운 값들을 찾는다.

만약 초기값이 잘 배열된다면 가까운 값들을 찾는 계산 시간과 비용은 줄어들게 된다. 이를 위해 큰 진폭 값의 개수를 기준으로 평균화하여 결과값을 구한 후 다시 이전 보다 조금 적은 데이터의 개수를 기준으로 평균화 하는 식으로 여러 단계를 그림 3과 같이 진행하여 최종적으로 결과물을 구한다.

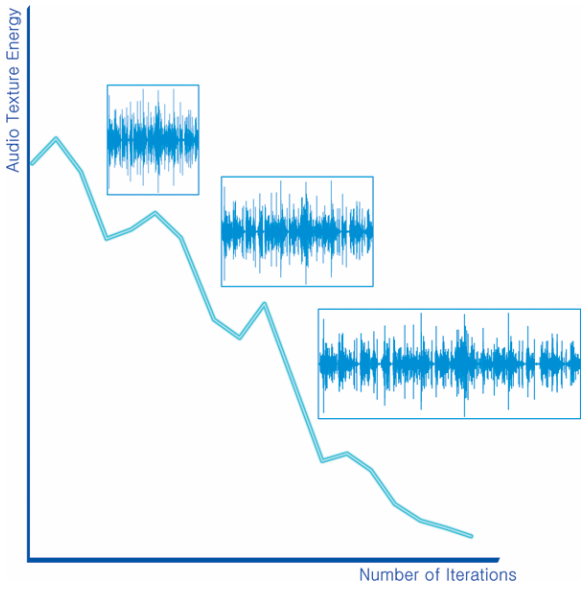


그림 3 다단계 합성 과정

5. 제약 조건을 포함한 합성

제약 조건이 없는 경우, 합성은 예제 오디오 클립과 임의의 재생 시간을 가지고 원본과 유사한 오디오 클립을 만들어 주는 것을 말한다.

이에 더하여 우리는 EM 최적화 과정에 제약 조건을 부여하여 예제 오디오 클립의 손실된 부분을 복구하거나 혹은 오디오의 특정 위치에서 사용자가 원하는 부분의 음악이 재생될 수 있게 하고 이로써 특정 흐름을 만들어낼 수 있게 하였다.

5.1 손실된 부분 복구

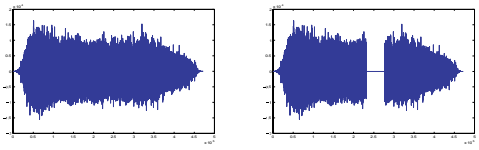
손실된 오디오 클립의 복구는 사용자에 의해 주어진 제약 조건을 적용시킨 EM 최적화 과정을 통해 간단히 복구될 수 있다.

복구 시 제약 조건은 두 가지로 축약될 수 있다. 첫 번째는 손실된 음악을 입력으로 줄 때, 손실 부분은 이웃으로 가져오면 안 된다는 제약 조건을 주어야 한다. 이는 이웃들을 탐색할 때 전체 탐색을 통해서 가져오기 때문에 만약 손실 부분에 대한 배제 없이 이웃들을 탐색 시 손실된 부분에 가

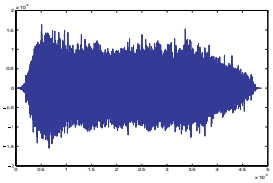
장 잘 매칭이 잘 되는 부분이 원래 손실된 부분이 되는 문제가 생길 수 있다. 그렇기 때문에 손실 부위를 배제한 상태에서 나머지 부분의 데이터를 이웃으로 가져와야 한다.

다음 제약 조건은 손실된 부분과 그 주위의 임의의 부분을 제외한 나머지 부분들을 고정시켜야 한다는 것이다. 손실된 부위를 동시에 복원될 결과로 가정해야 하는데 손실 부위를 제외한 나머지 부분은 고정시켜야 한다. 고정 시키지 않는다면 합성 시에 역시 전체 탐색을 통해서 원본 데이터의 구조가 바뀌는 현상이 있을 수 있기 때문이다. 또한 추가적으로 손실 부위의 주변 값들은 고정 시키지 말고 겹쳐질 수 있는 정도의 크기로 놔둔다. 겹쳐지는 부분이 없을 경우 피치가 끊어지는 문제가 발생하여 비정상적인 음을 들려줄 수 있다. 그렇기 때문에 일정 부분이 겹쳐져야 보다 자연스러운 결과를 들려 줄 수 있다.

이런 제약 조건을 주게 되면 결국 손실되지 않은 부분에서 가장 적합한 이음새를 지닌 부분이 추출되고, 그 부분을 메워 주게 되어 손실 부위를 복구해준다(그림 4). 초기에 손실 부위에는 임의의 값들을 넣어줌으로써 보다 자연스럽게 합성되게 할 수 있다.



(a) 원본 오디오 텍스처 b) 일부가 손실된 오디오 텍스처



(c) 복구된 오디오 텍스처

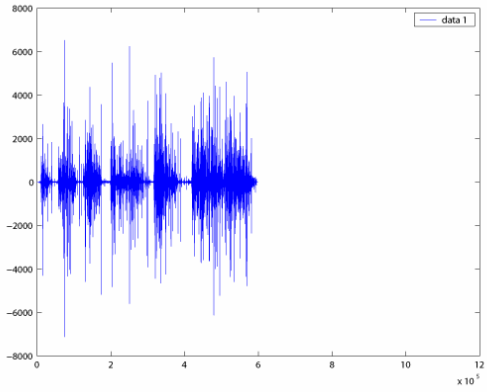
그림 4 복구된 오디오 텍스처와 원본과의 비교

5.2 음의 흐름 유도

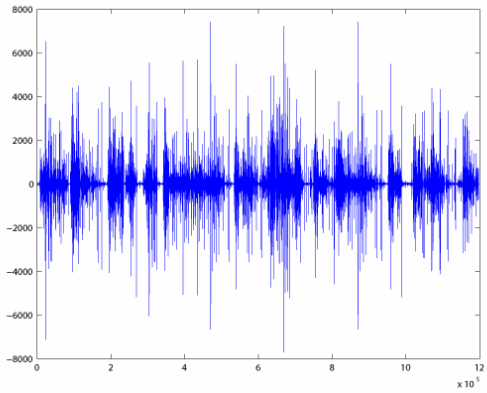
우리가 제안한 방법을 활용하면 예제 오디오 텍스처를 이용하여 사용자가 원하는 흐름의 형태로 오디오가 재생될 수 있게 만들어 준다. 예를 들어 예제 오디오 텍스처 안에 비가 천천히 내리거나 혹은 많이 내릴 때의 소리가 포함되어 있다면 이를 이용하여 빗소리가 조금씩 비가 내리다가 점점 많이 오게 하는 비의 흐름을 유도할 수 있다. 반대로도 역시 가능하고 점점 천천히 혹은 조금 내렸다 다시 많이 내리고 다시 조금 내리거나 하는 식의 복잡하고 다양한 형태로도 만들 수 있다.

이런 접근은 역시 초기값을 정할 때 제약 조건을 줌으로써 유도해 낼 수 있다. 가령 천천히 비가 내리다 점점 많이 내리는 예제 오디오 클립이 있다고 가정하고 합성된 결과는 예제 오디오 클립과 역으로 비가 많이 내리는 소리를 점점 천천히 비가 내리는 소리로 만들고 싶다면 유저는 예제의 비가 많이 내리는 부분을 결과의 초기값에 앞부분에 두고 고정시키고 조금씩 내리는 빗소리는 뒷부분에 배치하고 고정시키고 나머지 부분에 대해 앞서 제시한 알고리즘을 그대로 적용시켜 합성을 하게 되면 거친 빗소리에서 천천히 내리는 빗소리를 들려줄 수 있는 데이터가 생성된다. 물론 고정시키는 부분들의 거리가 멀게 되면 고정시키는 부분들 사이에 있는 부분들은 랜덤 혹은 적절한 값으로 바뀌게 된다. 또한 피치가 끊기는 문제로 인해 고정시키는 부분들을 너무 가까이 두면 듣기 좋지 않은 결과를 초래할 수 있다.

6. 결과의 비교 및 분석



(a) 원본 오디오 텍스처



(b) 결과 오디오 텍스처

그림 5 예제 오디오 텍스처와 결과 오디오 텍스처

그림 5는 원본 오디오 텍스처(a)와 이를 기반으로 합성시켜 약 2배 길이를 가지게 된 결과 오디오 텍스처(b)를 보여준다. (a)와 (b)를 비교해보면 그래프상에서 비슷한 형태의 구간이 존재하지만 일정 부분이 반복되거나 혹은 전체적으로 비슷한 느낌이 지속되는 모습을 보여주지 않기 때문에 반복되는 느낌을 주지 않을 수 있다.

과연 이 합성 방법이 올바른 방법이고 기존 방법과 어떤 점이 다른 가에 대해 알아보기 위해 일반인 8명을 대상으로 설문조사를 행하였다. 이들은 오디오에 대한 지식이 풍부한 전문가들이 아니며, 실생활에서 합성 결과를 듣게 될 대상자들은 일반인이 대부분이라고 예측할 수 있기 때문에 비전문가인 일반인을 대상으로 설문 조사하였다. 설문 조사는 오디오에 대해 다양한 패턴을 평가해주는 다양성, 얼마나 원본 오디오 예제와 비슷한 느낌이 나는가에 대한 자연스러움, 그

리고 합성 자체에 있어서의 질에 대한 평가와 같이 세가지 항목으로 구성되었다.

본 논문의 결과를 Lie Lu의 'Audio Textures'의 결과와 직접 비교를 해보았다. 즉 Lie Lu의 논문에서 사용한 사운드 샘플을 그대로 사용하여 결과를 만든 후, 위의 피실험자들에게 들려주었다. 표 1은 각각의 평가 항목에 대한 그래프이다. 각각의 평가 항목은 3등급으로 나누어 점수를 매겼고 항목에 대한 총합을 내었다.

<표 1> EM 최적화를 이용한 합성의 결과와 Audio Textures 의 결과 비교 평가

	Audio Textures	**EM 최적화**
자연스러움	56	**58**
다양성	52	**61**
결과의 질	55	**59**
총합	163	**178**

표 1 에서 조사 결과를 전체적으로 살펴보면 대체로 우리가 제안한 접근 방법(EM 최적화)이 기존 방법(Audio Textures) 보다 나은 결과를 들려준다고 평가되었다. 각각의 항목을 분석하여보면, 다양성의 경우 기존의 방법은 반복성이 쉬이 느껴지지만 이와 비교하여 우리의 방법은 반복되는 느낌을 덜 느낄 수 있었다고 평가되었다. 합성 후 원본과의 유사성을 평가하는 자연스러움도 마찬가지로 좋은 평가를 보여주었고, 사용자들에게 얼만큼 받아들여질 수 있는 가에 대한 평가인 합성 결과의 질에 대해서도 역시 좋은 평가를 보여주었다.

이 결과를 분석하여 보면 우리가 제안한 논문의 특징을 알 수 있다. 우선 자연스러움에 대한 비교는 그다지 차이가 나지 않았다. 셋 항목 중 가장 주관적인 영향이 미칠 수 있는 항목임에도 불구하고 많이 차이가 나지 않았다. 이는 다른 방법에 비하여 원본을 크게 벗어나지 않은 오디오를 합성할 수 있다는 것을 말해준다. 결과의 질의 경우 이전 방법보다 나은 결과를 들려주거나 이전의 방법처럼 사용자들에게 받아들여질 수 있다는 것을 의미한다. 결국 설문 조사를 통해서 우리가 제안한 방법 중에 'Audio Textures'와 차이점을 알 수 있는 것은 다양성에 관한 것이었다. 자연스러움에 대한 평가는 그다지 차이가 많이 나지 않았지만 다양성은 차이가 꽤 있었다. 이는 우리의 방법이 그래프 간의 확률을 기반으로 한 조각들에 의해 얻어진 그래프를 탐색하는 것 보다 반복성이 낮은 오디오를 들려준다는 의미로 해석될 수 있다. 또한 조각들간에 보다 자연스러운 연결성을 보장한다고 볼 수 있다.

7. 한계점

본 논문에서 제시한 방법은 짧은 패턴이 반복되거나, 짧은 사운드에서는 좋은 결과를 보여주지만 긴 패턴이 나타나는 구조를 가진 오디오 클립에는 적합하지 않다는 단점이 있다. 그 이유는 각 이웃을 설정하고 비교하는 단위가 짧으므로 최적화 방법을 사용하더라도 긴 구조를 모두 유지하기에는 힘들기 때문이다. 더불어 오디오 신호의 길이가 길어질수록 계산량이 많아지기 때문에 계산 시간이 길어진다는 단점도 있다.

8. 결론 및 향후 과제

본 논문에서는 오디오 텍스처 합성에 대한 새로운 접근으로 EM 최적화를 이용한 오디오 텍스처 합성 방법을 제안했다. 이는 예제 오디오 클립을 일정 단위로 나누고 이렇게 나눠진 부분들을 서로 일부가 겹쳐지게 합성하여 임의의 길이의 오디오 클립을 만든다. 그 후 만들어진 오디오 클립을 예제 오디오 클립과 부분 부분을 비교하여 확장된 오디오 클립과 최대한 비슷한 부분을 예제 오디오 클립에서 찾는다. 찾아진 결과를 결과 오디오에 다시 합성하여 오디오 클립을 만든다. 이런 과정을 반복하여 최적화된 가장 적절한 결과값을 구한다. 이 결과는 분할된 부분들이 가장 자연스럽게 이어지는 결과가 된다.

일반적으로 합성 결과의 관점에서 생각해봤을 때 한정적인 예제 오디오 클립 내의 데이터 수로 인해 이것을 확장한다면 단순히 반복되는 형태가 될 것이라고 짐작할 것이다. 하지만 본 논문에서 제안한 방법으로 합성된 결과는 분할된 이웃간에 자연스럽게 이어질 뿐 아니라 초기값 혹은 제약 조건에 따라 다양한 결과를 들려줄 수 있는 결과를 유도해 낼 수 있다.

본 논문은 'Audio Textures' 논문과는 달리 오디오를 직접 합성함으로써 단순히 그래프 이동을 통해 얻어지게 되는 반복성을 줄이면서도 예제 오디오 텍스처와 비슷한 느낌을 갖게 해주는 결과 오디오 텍스처를 생성할 수 있는 방법을 제안하였다.

본 논문에서는 정확한 합성을 위해 EM 최적화를 적용하였다. 추가적으로 최적화 과정에 몇몇 제약 조건을 적용함으로써 손실된 부분을 복구하거나 사용자의 요구대로 음의 흐름을 유도할 수 있게 하였고 이웃의 크기와 겹쳐지는 크기를 조절함으로써 다양한 결과를 들려줄 수 있게 하였다.

향후 가능하다면 오디오의 구조를 보다 더 잘 보존할 수 있고 특징까지 얻을 수 있는 방법에 대한 연구가 필요할 것이다. 만약 박자나 오디오에 관한 기타 여러 가지 정보가 주어질 때의 합성 방법은 어떤 식으로 접근할 수 있을지도 고려해봐야 할 것이다. 또한 계산과 비용의 측면에 있어서 보다 빠른 방법으로 합성하는 접근할 수 있는 방법을 연구해야 할 것이다.

참고문헌

[1] Perry R. Cook. Real Sound *Synthesis for Interactive Applications*. A.K.Peters:Natick, MA, 2002.

[2] A. Zils and F. Pachet, "Musical mosaicing," in *Proc. Cost G-6 Conf. Digital Audio Effects DAFX-01*, Limerick, Ireland, 2001.

[3] Lie Lu, Liu Wenyin, and Hong-Jiang Zhang. Audio textures:Theory and applications. IEEE Transactions on Speech and Audio Processing, Vol.15 No.2 pp. 156-167, 2004.

[4] Arno Schödl, Richard Szeliski, David H. Salesin, and Irfan Essa. Video textures. In Kurt Akeley, editor, *Siggraph 2000, Computer Graphics Proceedings*, pages 489-498. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000.

[5] J. Foote, "Visualizing music and audio using self-similarity," in *Proc. ACM Multimedia '99*, Orlando, Florida, November 1999, pp. 77-80.

[6] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *ACM Trans. Graph.*, 24(3):795-802, 2005.

[7] G. McLachlan and Krishnan T. The EM algorithm and extensions. 1997.

[8] Li-Yi Wei, and Marc Levoy, M. 2000. Fast texture synthesis using tree-structured vector quantization. *Proceedings of Siggraph 2000,* July, 479-488.

[9] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra., 2003. Graphcut textures: Image and video synthesis using graph cuts. *ACM Transactions on Graphics, Siggraph 2003* 22, 3, July, 277-286.

[10] Jeremy S. De Bonet. Multiresolution sampling procedure for analysis and synthesis of texture images. *Computer Graphics*, 1(Annual Conference Series):361-368, 1997.