# 마르코프 체인과 계층적 클러스터링 기법을 이용한 작곡 기법

## Music Composition Using Markov Chain and Hierarchical Clustering

권지용, Ji-yong Kwon*, 이인권, In-Kwon Lee**

**요약** 본 논문에서는 주어진 예제 멜로디 데이터를 이용하여 효과적으로 새로운 곡을 작곡하는 시스템을 제안한다. 우리가 제안하는 기법은 k-차원 마르코프 체인을 이용하여 마디 단위의 음악 블록을 합성한다. 한마디 단위를 하나의 마르코프 체인의 상태로 취급할 경우 매우 많은 상태를 고려해야 하므로, 이를 계층적 클러스터링 기법을 통하여 학습이 용이한 정도로 상태를 줄인다. 예제 데이터의 각 음악 블록은 소속된 클러스터 번호의 시퀀스로 대체되어 학습 데이터로 사용된다. 학습된 마르코프 체인의 상태를 전이하면서 각 상태에 해당되는 클러스터의 음악 블록을 랜덤하게 선택하여 합성한다. 학습된 마르코프 체인은 효과적으로 예제 음악과 비슷하면서 새로운 곡을 생성할 수 있었다.

**Abstract** In this paper, we propose a novel technique that generate a new song with given example songs. Our system use k-th order Markov chain of which each state represents notes in a measure. Because we have to consider very high-dimensional space if we use notes in a measure as a state of Markov chain directly, we exploit a hierarchical clustering technique for given example songs to use each cluster as a state. Each given examples can be represented as sequences of cluster ID, and we use them for training data of the Markov chain. The resulting Markov chain effectively gives new song similar to given examples.

***핵심어: 마르코프 체인, 알고리즘 작곡, 부동성 계산, 계층적 클러스터링 (Markov Chain, Algorithmic composition, Dissimilarity computation, Hierarchical clustering)***

본 논문은 문화관광부 및 한국문화콘텐츠진흥원의 문화콘텐츠기술연구소(CT)의 육성 사업의 연구 결과로 수행되었음.

*주저자 : 연세대학교 컴퓨터과학과 석박사통합과정 e-mail: mage@cs.yonsei.ac.kr

**교신저자 : 연세대학교 컴퓨터과학과 교수; e-mail: iklee@yonsei.ac.kr

## 1. 서론

마코프 체인(Markov chain)은 컴퓨터 과학 여러 분야에서 매우 다양한 용도로 활용되고 있는 상태 기계(state machine)의 일종으로서, 컴퓨터를 이용한 작곡법의 용도로서도 이미 오래 전부터 사용되어 왔다[1]. 마코프 체인은 예제 데이터 기반의 학습이 용이한 구조이므로 적절한 예제 데이터를 이용하여 마코프 체인을 학습하면 이론적으로 비슷한 형태의 데이터 나열을 생성하거나 입력된 데이터 나열의 식별 등에 이용할 수 있다. 따라서 특정 형식의 여러 곡을 예제 데이터로 하여 마코프 체인을 학습시키면 예제 데이터와 유사한 여러 곡을 생성할 수 있을 것이다. 한 예로 Verbeurt 등[4]은 접미 트리(suffix tree) 구조를 사용하여 음악의 패턴을 표현하고 이를 마코프 체인을 통해 학습하여 유사한 패턴을 찾아내는 방법을 제안하고, 이러한 음악 패턴을 자동 작곡 기법에 응용할 수 있음을 보였다.

과거에 연구된 마코프 체인을 이용한 작곡법들은 대체적으로 하나의 상태(state)가 하나의 음을 나타내고, 시간적으로 마코프 체인의 상태가 확률적으로 변화하면서 각각의 상태를 나타내는 음이 나열되어 작곡이 되는 형태를 띠고 있다. 이러한 형태의 마코프 체인은 상태의 개수가 비교적 적기 때문에 기계 학습 기법(machine learning)이 효과적으로 적용이 될 수 있다. 그러나 개별적인 음을 하나의 상태로 나타낼 경우 전체적인 음악의 구조나 화성의 고려를 하기 힘들다. 대안으로서 마코프 체인의 차수(order)를 늘리는 등의 해결법이 있으나, 차수를 하나씩 늘릴 때마다 고려해야 하는 상태의 개수가 지수배로 늘어나기 때문에 처리에 한계가 있다. 상태를 하나의 음이 아닌 마디 단위나 그와 비슷한 음의 묶음 단위로 나누어 볼 수도 있을 것이다. 이러한 형태로 마코프 체인을 구성할 경우, 음악의 구조적 특성이나 화성의 고려가 앞에서 언급한 기법에 비해 훨씬 수월해지게 된다. 그러나 다수개의 음이나 마디 단위로 하나의 상태를 표현하게 될 경우 고려해야 하는 상태의 개수가 거의 무한개에 가까울 정도로 많아지게 되며, 따라서 기계 학습 기법을 적용하기 어려워진다.

[6]에 소개된 음악 그래프(Music graph) 기법은 주어진 여러 개의 음악을 입력으로 하여 각 음악을 마디단위로 나누어 각각을 그래프의 버텍스로 두고, 두 유사한 마디의 다음 마디를 엇갈린 에지로 연결한 그래프를 생성한 뒤, 이 그래프를 선회하며 얻어진 패스를 이용한 새로운 음악을 만들어 내는 방식으로써 좋은 결과를 보여주었다. 그러나 그래프의 구조가 매우 복잡해질 가능성이 있으며, 이를테면 16 마디 단위의 소절 규칙을 지키는 패스를 구하는 등의 특정한 구조적 규약을 지키기 힘들다.

우리는 위와 같은 문제를 해결하기 위해 음의 마디 단위를 마코프 체인에서의 상태로 사용하기 위해서 효율적으로 상태의 개수를 줄이는 방법을 제안하고, 이를 통해 구성한 마코프 체인을 학습하여 새로운 음악을 생성하였다. 음의 마디 단위를 하나의 상태로 나타내기 위한 핵심 아이디어는 계층적 클러스터링(hierarchical clustering) 기법을 통해 비슷한 마디를 같은 상태로 묶어 표현하고, 이렇게 묶여진 각각의 클러스터를 마코프 체인의 상태로 사용하는 것이다.

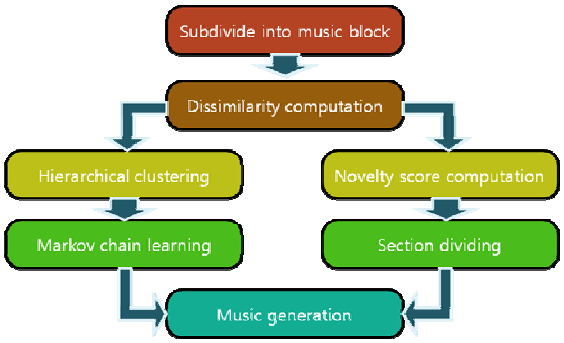


**그림 1. 제안하는 방법의 전체 흐름도**

## 2. 제안하는 방법

우리가 제안하는 방법은 그림 1 과 같이 구성된다. 예제로 사용되는 데이터는 타악기 체널(대체적으로 10 번)이 배제된 미디 데이터를 기본으로 한다. 먼저 예제 데이터를 마디 단위로 나눈다. 이렇게 마디 단위로 나누어진 멜로디 데이터를 '음악 블록' 이라고 칭하겠다. 다음 모든 블록에 대하여 다른 음악 블록간의 부동성(dissimilarity)을 분석한다. 두 블록간의 부동성을 저장한 부동성 행렬(dissimilarity matrix)을 계산하면 이를 이용하여 계층적 클러스터링 기법을 적용할 수 있다. 원본 음악 블록의 개수보다 적은 수의 클러스터를 만들게 되면 각각의 클러스터를 하나의 상태로 하는 마코프 체인을 생성하고 예제 데이터를 사용하여 학습한다. 동시에 음악의 구조적 특성을 알아내기 위해 부동성 행렬을 이용하여 노벨티 점수(Novelty score, [8])를 계산한 다음, 이를 이용하여 각 음악 블록마다 소절 번호를 할당한다. 마지막으로 학습된 마코프 체인과 음악 블록들의 소절 번호를 이용하여 새로운 음악을 생성하게 된다.

### 2.1 부동성 계산

두 음악 블록간의 부동성을 계산하는 가장 쉬운 방법은 각 음악 블록을 수치 데이터로 나타낸 뒤 둘의 유클리드 거리(Euclidean distance)를 사용하는 것이다. 그러나 이 방법은 음악적으로 두 음악 블록의 유사성 및 부동성을 고려하지 못할 가능성이 크다. 따라서 본 논문에서는 두 음악 블록간의 부동성을 효과적으로 계산하는 방법을 제안한다.

두 음악 블록 간의 부동성을 계산하기 위해서는 두 음표 간의 부동성 계산 방법을 효과적으로 설계해야 한다. $i$ 번째 음악 블록의 $p$ 번째 음표를 $n_{i,p}$, $j$ 번 음악 블록의 $q$ 번째 음표를 $n_{j,q}$ 라고 했을때, 우리가 제안하는 두 음표 간의 부동성 계산 방법은 다음과 같은 식으로 나타낼 수 있다.

$$D(n_{i,p}, n_{j,q}) = w_{pit} d_{pit}(n_{i,p}, n_{j,q}) + w_{oct} d_{oct}(n_{i,p}, n_{j,q}) + w_{dur} d_{dur}(n_{i,p}, n_{j,q}) + w_{time} d_{time}(n_{i,p}, n_{j,q})$$

여기서 $d_{pit}$, $d_{oct}$, $d_{dur}$, 그리고 $d_{time}$ 은 각각 음정의 차이값, 옥타브의 차이값, 음길이의 차이값, 그리고 음의 시작 시간의 차이값을 나타내는 것으로 각기 다른 형태의 계산식을 가지고 있다. 이들의 가중합을 통하여 두 음표 간의 부동성을 계산하게 된다. 먼저 음정의 차이값을 계산하는 방법된 [5]에 소개된 음정 불안정성(Interval instability) 표를 이용하였다. 즉, 가장 작은 불안정성을 가지고 있는 완전 1 도를 0.0 으로, 가장 큰 불안정성을 가지고 있는 감 2 도를 1.0 으로 보고 이를 선형적으로 등분한 값을 음정의 차이값으로 사용하게 된다. 옥타브의 차이값, 음길이의 차이값, 그리고 음의 시작 시간의 차이값을 이용한 부동성 계산식은 다음과 같다.

$$d_{oct}(n_{i,p}, n_{j,q}) = \lfloor |p(n_{i,p}) - p(n_{j,q})| / 12 \rfloor$$

$$d_{dur}(n_{i,p}, n_{j,q}) = \left| \log_2 \frac{l(n_{i,p})}{l(n_{j,q})} \right|$$

$$d_{time}(n_{i,p}, n_{j,q}) = \left( |t(n_{i,p}) - t(n_{j,q})| / T \right)$$

위 식에서 p(n), l(n), t(n)은 각각 음표 n 의 음높이, 음길이, 마디의 첫 시작 시간을 기준으로 한 음의 시작 시간을 말한다. 옥타브의 차이값은 두 음이 한 옥타브 이상 차이가 날 때마다 1.0 씩 증가하는 형태로 부동성을 계산한다. 음길이의 차이값은 두 음의 길이의 비를 이용하여 계산하며 2 배 차이가 날 경우 1.0 씩 증가하게 된다. 음의 시작 시간의 차이는 한 박 이상 차이가 날 경우 1.0 씩 증가하도록 식을 설계하였다. 각 부분의 가중치는 차후에 소개되는 계층적 클러스터링의 결과가 좋은 형태로 나타날 때까지 수동적으로 조정하였다. 우리가 실험에서 사용한 가중치 값은 각각 $w_{pit}$=5.0, $w_{dur}$=0.5, $w_{oct}$=1.0, $w_{time}$= 0.5 이었다.

두 음표 간의 부동성 계산 방법을 기초로 하여 다음과 같은 방법으로 두 음악 블록 간의 부동성을 계산한다. 두 개의 음악 블록을 각각 $B_i$, $B_j$ 라고 하였을 때, 기본적인 부동성 계산 방법의 아이디어는 음악 블록 $B_i$ 에 속한 음표 $n_{i,p}$ 에 대하여 이와 가장 잘 대응되는 음악 블록 $B_j$ 에 속한 음표 $n_{j,q}$ 를 찾고 이렇게 각 매칭되는 음표 간의 부동성 점수의 평균을 사용하는 것이다. 이를 식으로 표현하면 다음과 같다.

$$C(n_{i,p}) = \min_q D(n_{i,p}, n_{j,q})$$

$$M(n_{i,p}) = \arg\min_q D(n_{i,p}, n_{j,q})$$

$$\mathbf{D}(B_i, B_j) = \sum_p C(n_{i,p}) \Big/ N(B_i)$$

음악 블록 $B_i$의 음표와 대응되는 음악 블록 $B_j$의 음표를 찾기 위해 우리는 소진 검색(exhaustive search)을 이용하였다. 때문에 음악 블록 $B_i$의 음표 개수가 p 개이고, 음악 블록 $B_j$ 의 음표 개수가 q 개라면, 부동성 계산 알고리즘의 복잡도는 O(pq)가 되며, 한 블록의 음표의 평균 개수가 p 개라면 복잡도는 $O(p^2)$으로 볼 수 있다. 그러나 일반적인 음악 데이터에서는 대체적으로 한 블록에 속한 음표의 개수가 그리 많지 않으므로, 소진 검색을 사용하더라도 빠른 시간 내에 계산이 가능하다.

그림 2 는 우리가 제안한 부동성 계산 방법을 통해 두 개의 실험 음악의 각 음악 블록간 부동성 계산을 하여 부동성 행렬을 생성한 것을 그레이 스케일 이미지(Grayscale image)로 나타낸 것이다. 검은색일수록 두 대응되는 음악 블록이 비슷함을 의미하며, 흰색일수록 두 대응되는 음악 블록이 다름을 의미한다. 그림에서 같은 흑백 패턴이 계속 반복되는 것은 소절 내부에서의 코드(chord) 변경에 관련된 현상이며, 즉 비슷한 코드를

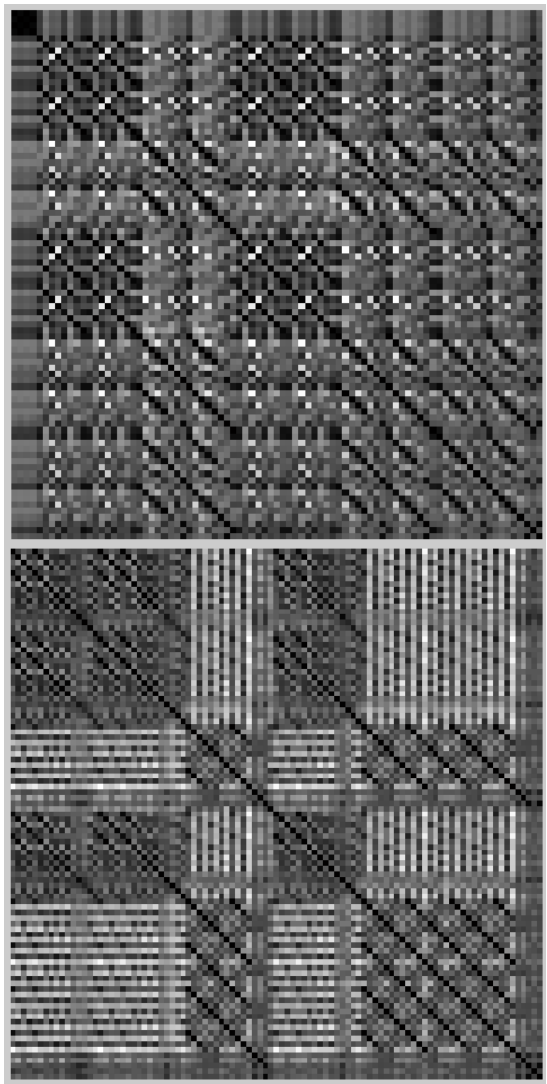

**그림 2. 부동성 계산 방법을 이용하여 부동성 행렬을 구성한 예**

가진 음악 블록은 적은 부동성 값을 갖게 됨을 의미한다.

또한 전체적인 패턴이 현격한 차이를 보이는 것을 통해서 각 부분이 서로 다른 소절을 구성하고 있음을 알 수 있다. 이와 같은 사실에서 알 수 있듯이, 우리가 제안한 부동성 계산 방법은 음악 블록 간의 유사성을 잘 표현할 수 있으며 또한 전체적인 소절의 유사성까지 고려할 수 있음을 확인할 수 있다.

## 2.2 계층적 클러스터링

앞의 부동성 계산 방법을 통해 각 음악 블록의 쌍마다의 부동성을 계산하면 부동성 행렬을 계산할 수 있다. 부동성 행렬을 이용하여 계층적 클러스터링 기법을 사용해 음악 블록들을 정해진 개수 m 개의 클러스터로 나눈다. 우리가 사용한 클러스터링 기법은 응집 계층적 클러스터링 기법(agglomerative hierarchical clustering)으로서 기본적으로 각 샘플간의 부동성 값만 가지고 있으면 클러스터링을 수행할 수 있다. 일반적으로 많이 사용되는 분할 클러스터링 기법(partition clustering)은 기법의 특성상 클러스터링의 대상이 되는 샘플들의 중심점(centroid)를 계산할 수 있어야 하는데, 우리가 고안한 부동성 계산 기법은 하나의 샘플을 벡터화하여 나타내는 방식이 아니기 때문에 적용이 힘들고 또한 계산된 부동성도 유클리드 공간의 특성을 만족하지 못하기 때문에 적합하지 못하다. 우리는 두 개의 클러스터 간의 부동성 계산을 위하여 다양한 방법을 선택하여 실험한 결과, 일반적으로 사용되는 single 기법에 비해 각 클러스터가 고른 개수를 가지게 되는 complete 기법을 사용하였다.

우리의 방법에서 이용하는 마코프 체인은 클러스터의 총 개수를 통해 상태 개수를 결정하므로, 적당한 클러스터의 개수를 정하는 작업이 매우 중요하다. 만일 적당한 클러스터의 개수보다 더 적은 클러스터의 개수를 사용할 경우, 학습 시퀀스가 비슷한 모양을 하게 되므로 학습 데이터가 나쁜 모양을 하게 되며, 또 적당한 클러스터의 개수보다 더 많은 수의 클러스터를 사용하게 되면, 유일한 학습 시퀀스가 많아져 학습 데이터의 절대량이 부족하게 된다. 우리는 반복적인 실험을 통해서 컷 오프 상수를 약 1.153 으로 정하면 대체적으로 적당한 수의 클러스터를 자동적으로 생성하는 것을 확인하였다.

## 2.3 노벨티 점수를 이용한 소절 구분

앞의 클러스터링 기법을 이용하여 묶여진 각 클러스터를 하나의 상태로 두고 마코프 체인을 학습한 뒤, 마코프 체인을 통해 새로운 상태 시퀀스를 생성하고 생성된 시퀀스에 각 상태마다 해당하는 클러스터의 음악 블록을 랜덤하게 결정하면 새로운 곡을 생성할 수 있다. 그러나 이와 같은 방법만으로는 음악에서 흔히 찾아볼 수 있는 소절의 규약을 보장하기 힘들어진다. 즉, 첫 소절에 나오는 음악 블록이 나오다가 갑작스럽게 클라이막스 부분의 음악 블록이 나올수도 있고, 클라이막스 부분에서 갑작스럽게 첫 중간 소절의 음악 블록이 나올 수도 있게 된다. 이는

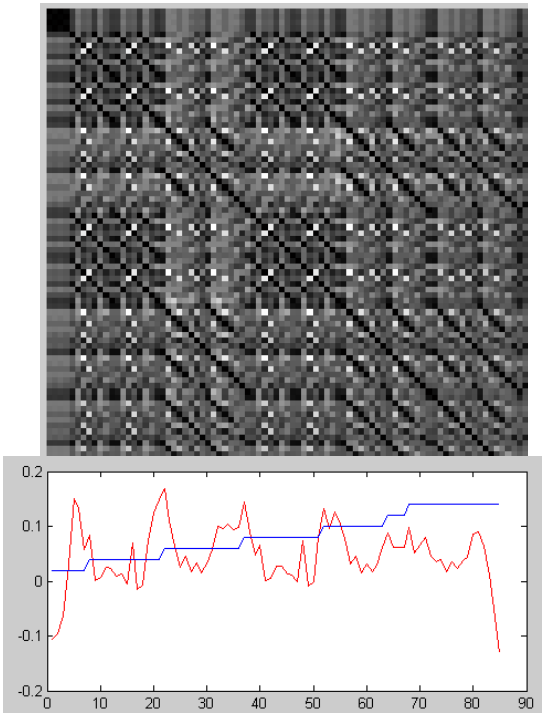


**그림 3. 노벨티 점수를 이용한 음악 소절의 구분의 예**

음악을 구성하는 각 소절이 악기 구성이나 연주법 등은 다르더라도, 같은 코드 패턴을 띄기 쉽기 때문에 일어나는 현상으로 볼 수 있다.

우리는 이와 같은 현상을 막기 위해 주어진 음악의 소절을 자동적으로 구분하여 이를 새로운 음악을 생성하는데 사용하였다. 자동적으로 음악의 소절을 구분하기 위해서 우리는 오디오 신호의 소절 구분을 위해 자주 사용되는 노벨티 점수[8]를 이용하였다. 일반적으로 노벨티 점수는 오디오 신호를 적당한 길이 간격으로 나눈 다음 각 구간을 주파수 범주의 벡터로 표현하여 이들간의 차이를 이용하지만, 결국 부동성 행렬만 생성되면 어떤 형태의 데이터이든지 노벨티 점수를 적용할 수 있다는 점을 이용하였다. 그림 3 은 주어진 음악의 부동성 행렬을 계산하고, 여기에 각 노벨티 점수를 계산하고 이 점수의 지역적 최대점을 추출하여 음악 소절을 구분한 것이다. 지역적 최대점을 추출하는 방법은 전통적인 방법 대신 노이즈에 강한 평균 커브 방법[9]을 이용하였다. 그림을 보아 알 수 있듯이, 부동성 행렬과 노벨티 점수를 이용하여 각 소절을 효과적으로 구분하는 것을 확인할 수 있다.

## 2.4 마르코프 체인 학습과 결과 생성

앞의 클러스터링 과정을 거치면 마지막으로 하나의 클러스터를 마르코프 체인의 상태로 두고 마르코프 체인을 학습한다. 앞의 과정을 통해 예제 음악 데이터를 이루는 각 음악 블록이 어떤 클러스터에 속해있는지 정해지므로, 이를 이용하여 예제 음악 데이터의 상태 시퀀스(sequence)를 얻어올 수 있다. 마르코프 체인의 각 상태에서 다른 상태로의 전이 확률을 랜덤한 값으로 초기화한 후, 앞서

얻어온 상태 시퀀스를 반복적으로 학습하게 한다. 학습은 크게 두 단계를 반복함으로써 가능해지는데, k 개의 상태를 보고 다음 상태로 전이하는 마르코프 체인의 경우 상태 시퀀스의 입력 상태 k 개와 출력 상태 1 개를 전이 확률값이 저장된 표에서 가져와 이를 일정 배수 증폭시키고 다른 확률값을 감소시킨 뒤, 몇 개의 예제 시퀀스에 대해 테스트를 수행하여 성능을 테스트한다. 상태의 전이가 예제와 같이 올바르게 동작할 확률이 t 에 도달할 때 까지 계속해서 학습을 한다. 실험에서 우리는 k 를 3, t 를 0.95 로 주었다.

새로운 음악을 생성하기 위해서 우리는 다음과 같은 수행을 하였다. 먼저 마르코프 체인의 k 개의 상태를 예제로부터 얻어와서 초기화한 다음, 현재 소절 번호를 기억해둔다. 마르코프 체인의 다음 상태를 결정하면서 그 상태에 해당하는 클러스터에 속한 음악 블록 중 하나를 랜덤하게 선택하여 이어 붙여나가는 데, 이때 이전의 소절 번호와 같은 소절 번호를 가진 음악 블록이 있으면 우선적으로 선택하게 된다. 만일 미리 정해진 소절의 길이만큼의 음악 블록을 이어 붙였거나 클러스터 내부에 이전의 소절 번호와 같은 소절 번호를 가진 음악 블록이 없으면 클러스터 내부에 있는 음악 블록으로부터 새로운 소절 번호를 선택하게 된다. 이러한 방법을 사용하면 소절 규약을 어느 정도 유지할 수 있는 결과물을 생성할 수 있다.

## 3. 실험 및 결론

우리는 실험 데이터로 J.S.Bach 의 Brandenberg Concerto No.2 와 No.3 의 4 분의 4 박자 부분에 해당되는 멜로디 부분을 사용하였다. 이 곡들은 당김음이 없고 리듬이 비교적 고르기 때문에 음악 블록을 나누어 분석하는데 있어 추가적으로 고려해야 할 사항이 적으므로 우리의 실험 데이터로서 적합하였다. 실험에 사용된 멜로디의 전체 음악 블록의 개수는 253 개이며, 이를 클러스터링한 결과 21 개의 클러스터가 생성되었다. 실험적으로 생성된 500 마디의 노래는 원본 데이터를 음악적인 구조에 맞게 효과적으로 재구성한 것을 확인할 수 있었다. 또한 여러 소절의 음악 블록들이 뒤섞여 나오지 않고 한 소절 번호 내의 음악 블록들이 일정한 길이 이상으로 유지가 됨으로써, 소절 규칙을 어느 정도 유지하는 음악을 생성할 수 있었다.

우리가 제안한 방법은 기존의 방식으로 마르코프 체인을 사용할 경우 문제가 되었던 상태의 경우의 수를 효과적으로 줄여 결과적으로 구조적으로 강건한 음악을 생성할 수 있었다. 그러나 원본 데이터를 재구성한 형태의 결과를 생성하기 때문에 엄밀한 의미에서 작곡이라고 보기 힘들다는 한계점이 있다. 따라서 보다 창의적인 작곡을 위해서는 하나의 클러스터 안의 데이터를 예제로 비슷한 음악 블록을 생성하는 작곡 기계를 모델링하는 과제가 남아있다. 또한 보다 효과적인 클러스터링을 위해서 각 음악 블록간의 부동성을 계산하는 데에 있어서 가중치를 어떻게 조절하는지에 대한 문제나 클러스터의 개수를 효과적으로 제어하는 문제 등은 앞으로 계속하여 개선할 사항이다.

## 참고문헌


[1] Karlheinz Essl, "Algorithmic Composition," in Cambridge Companion to Electronic Music, Cambridge University Press, 2007.

[2] Min-Joon Yoo, In-Kwon Lee, "Musical Tension Curves and its Applications," In Proceeding of International Computer Music Conference 2006, pp.482-486, Nov.6-11. New Orleans, U.S, 2006.

[3] Earl Gose, Richard Johnsonbaugh, Steve Jost, "Pattern Recognition and Image Analysis," Prentice Hall PTR, 1996.

[4] Karsten Verbeurgt, Michael Dinolfo, and Mikhail Fayer, Extracting Patterns in Music for Composition via Markov Chains, IEA/AIE '2004: Proceedings of the 17th international conference on Innovations in applied artificial intelligence, pp.1123-1132, 2004.

[5] Min-Joon Yoo, and In-Kwon Lee, Musical Tension Curves and its Applications, In Proceeding of ICMC(International Computer Music Conference) 2006, pp.482-486, Nov.6-11. New Orleans, U.S., 2006.

[6] Hyun-Chul Lee and In-Kwon Lee, Synchronization of Background Music and Motion in Computer Animation, Eurographics 2005, Dublin, Ireland, Computer Graphics Forum 24(3) pp353~362 , Oct 31, 2005

[7] Min-Joon Yoo, In-Kwon Lee, and Jung-Ju Choi, Background Music Generation Using Music Texture Synthesis, ICEC(International Conference on Entertainment Computing) 2004, Technical University of Eindhoven, Eindhoven, The Netherlands. Lecture Notes in Computer Science, Volume 3166, pages 565-570, Sep 1st-3rd, 2004.

[8] Jonathan Foote, Automatic Audio Segmentation using a Measure of Audio Novelty, In Proceedings of IEEE International Conference on Multimedia and Expo, vol. I, pp. 452-455, 2000.

[9] Jackie Assa, Yaron Caspi, and Daniel Cohen-Or, Action synopsis: pose selection and illustration, SIGGRAPH '05: ACM SIGGRAPH 2005 Papers, pp.667-676, 2005