장면 보존적인 뮤직비디오 생성을 위한 다단계 분할 매칭 기법 *

윤종철°, 이인권†
연세대학교 컴퓨터과학과 비쥬얼컴퓨팅연구실
media19@cs.yonsei.ac.kr°, iklee@yonsei.ac.kr†

Scene Conserved Music Video Generation Using the Multi-Level Segmentation

Jong-Chul Yoon°,In-Kwon Lee†
Dept. of Computer Science, Yonsei University

요 약

뮤직 비디오란 주어진 음악과 비디오가 동기화 된 형태의 창작물을 뜻한다. 기존의 뮤직비디오 제작방식에서는 만들어진 음악을 위해 영상 촬영에 전문적인 촬영 기술을 요구하였다. 본 논문에선 보다 쉬운 뮤직비디오 생성을 위하여 비디오와 음악의 특성을 분석하여 자동적인 뮤직비디오 생성시스템을 소개한다. 두 개체의 연속성을 보장하는 비교를 위해 우리는 각각의 객체의 흐름을 분석하고, 흐름의 유사성을 기준으로 분할하는 기법을 제시한다. 분할된 영상과 음악의 특성 비교를 통한 최적화된 매칭기법 을 비롯하여, 보다 다양한 조각 생성을 위한 다중 레벨(multi-level)분할 기반의 매칭 기법을 소개한다. 본 논문의 기술을 사용하여, 일반인이 홈비디오 등을 사용하여 손쉽게 뮤직비디오를 제작할 수 있다.

1. 서 론

최근의 홈비디오 시장의 확장에 따라, 전문성이 배재된 비디오 편집기술의 요구되고 있다. 최근의 다양한 편집툴의 개발을 통해 일반적인 사람들도 비디오 클립의 인위적인 배치를 통한 비디오 편집 기능은 쉽게 사용할 수 있다. 하지만 뮤직비디오와 같이 영상과 음악이 연계성을 가지는 형태의 편집에 있어선 음악과 영상의 동기화를 만드는 전문적 기술이 필요하다. 기존의 뮤직비디오 제작방식은 미리 만들어진 음악에 동기화되기 위한 영상 촬영을 위하여 전문적인 촬영 기술을 요구하였다. 음악과 영상의 분리된 작업환경의 제약 때문에 수많은 시행착오를 요구하였고, 일반적인 홈비디오 사용자에겐 동기화된 뮤직비디오의 제작이 쉽지 않은 일이다. 따라서 본 논문은 전문적인 기술이 없는 일반 홈비디오 사용자가 손쉽게 영상과 음악을 자동적으로 연계시킬 수 있는 시스템을 소개한다.

본 논문의 목적은 비디오와 음악의 매칭을 통해 주어진 음악에 맞는 최적화된 비디오 클립을 추출하는 뮤직 비디오 자동 생성기법의 제시이다. 장면의 연속성을 보장하는 음악과 비디오의 매칭을 위해서 우리는 자동분할을 기준으로 한 매칭방법을 제시한다. 촬영자에게 있어서 하나의 비디오 샷 또는 클립은 특정한 연속적인 정보전달을 위해서 제작된다. 즉 비디오에 함축된 정보를 보호하기 위해선 촬영자에 의해 만들어진 이야기의 흐름을 깨지 않아야 한다. 따라서 본 논문에선 흐름을 측정하기 위한 유사성 측정 방법을 제안하고 이것의 분석을 통해 단계별 조각 매칭기법을 제시한다.

우리의 시스템은 크게 매체 분석 모듈과 매체 매칭모듈로 나눌 수 있다. 입력 영상은 일반적인 촬영물이고, 음악의 입력은 일반적인 웨이브파일을 기준으로 하였다. 데이터 분석 모듈에서는 음악과 비디오를 분할한 뒤 각 조각의 속도정보와 밝기정보를 분석하게 된다. 분석된 데이터베이스를 통해 주어진 음악의 조각에 가장 유사한 정보를 가지는 영상 조각을 찾아내는 작업을 매칭 모듈에서 해주게 된다. 만약 주어진 음악 조각에 맞는 적당한 영상을 찾지 못했을 경우, 다중 레벨(multi-level) 기반의 분할기법을 사용하여 음악 조각을 재분할해준 뒤 다시 매칭 되는 영상을 찾는다.

본 논문에서 제공하는 뮤직비디오 생성기법의 장점은 다음과 같이 세가지로 정리할 수 있다.

- **장면 연속성 보존:** 장면 보존을 위한 분할 단위의 매칭을 통해 비디오의 연속성을 최대한 유지할 수 있다.
- **인식기반 동기화:** 인식기반의 비디오 분석을 통해 시청자가 집중하는 부분의 동기성을 높여준다.
- **역동적 화면 생성:** 분할 단위의 매칭을 위한 타임 와핑(time-warping)을 통해 보다 역동적인 뮤직비디오의 생성이 가능하다.

본 논문의 구성은 다음과 같다. 2장에선 지금까지의 관련 연구를 밝힐것이고, 3장에서는 우리가 제안하는 시스템의 전체적인 흐름을 설명하겠다. 4장과 5장은 각기 영상과 음악을 다중 분할 하는 방법과 분석을 하는 방법을 제시할 것이고, 6장에서는 두 매체간의 매칭 기법에 대해 설명하겠다.

*본 연구는 ITRC(IT Research Center)산하 게임에니메이션 센터의 지원으로 수행되었음.

마지막으로 7장에서는 우리의 실험적 결과를 소개하고, 8장에서 결론및 향후과제를 밝히도록 하겠다.

2. 관련 연구

영상과 음악의 동기화에 대한 연구는 대부분 주어진 음악에 맞도록 영상 데이터를 수정 또는 재조합하는 방법으로 진행되었다. Foote [1] 등은 음악의 반복적인 특성을 이용하여, 음악의 유사행렬(similarity matrix)을 계산하고 여러 조각으로 나눴다. 또한 비디오의 밝기 변화와 카메라 움직임을 분석하여 비디오도 여러 조각으로 나눈 후, 각 조각의 변환점(transition point)을 맞추는 방법을 소개하였다. 비디오 조각 각각의 특징과 음악의 특징을 고려하는 방법은 Hua [2] 등에 의해서 시도되었다. Hua 등은 일반인이 찍은 홈비디오의 경우 화면의 질이 낮고 필요 없는 부분들이 많을 것이라는 전제를 바탕으로 물체의 동작, 카메라 동작, 오디오 등을 토대로 비디오 샷(shot)마다 집중도(attention score)를 계산하여 중요한 샷만을 요약하는 기법을 소개하였다. 선택된 비디오 샷들은 박자(beat)의 세기를 기준으로 나눠진 음악 조각들의 빠르기(tempo)에 맞도록 대응시킴으로써 동기화를 시도하였다. Mulhem [3] 등은 비디오 편집전문가들이 통상적으로 사용하는 몇 가지 미적인 규칙(aesthetic rules)들을 사용하여 음악의 변화에 적절한 내용(특성)의 비디오 조각을 붙여나가는 방법을 제시하였다.

앞에서 제시한 방법들은 비디오 조각의 재조합을 통한 동기화를 시도하였다면, Jehan [4] 은 주어진 비디오의 재생속도를 부분적으로 조정하면서 특징점의 동기화를 이루었다. 춤을 추는 비디오에 사용자가 직접 동작의 박자를 표시하고, 음악에서 분석된 템포를 맞추기 위해 비디오의 재생속도를 조정하는 방법이다.

본 연구에서 제시하는 비디오와 음악의 동기화 기법은 분할 매칭이라는 점에서 Foote 등의 기법과 유사하지만, 다중 레벨 기반의 조각 매칭을 통해 원본영상의 흐름을 최대한 유지한다는 점에서 장점을 가진다.

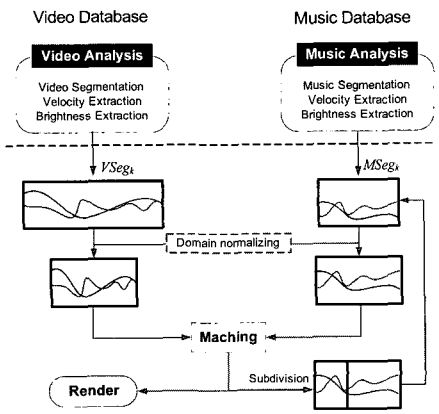

그림 1: 뮤직비디오 시스템 개요.

3. 뮤직비디오 시스템 개요

우리가 제안하는 시스템은 크게 데이터 분석 모듈과 데이터 매칭모듈로 나눌수 있다. 입력 영상은 일반적인 홈비디오 촬영물이고, 음악의 입력은 웨이브파일을 기준으로 하였다.

데이터 분석 모듈에서는 음악과 비디오를 분할한 뒤 각 조각의 특징을 나타내는 요소를 추출한다. 음악의 분할을 위해서 우리는 순간적인 주파수영역의 유사도를 기반으로 한 Novelty score [1] 를 사용하였다. 영상의 분할을 위해선, 영상 내의 형태의 급격한 차이를 기반으로 한 외형매칭(Contour shape matching) [5] 기반의 유사도 측정을 사용하였다. 각기 분할된 조각 단위로 속도 정보와 밝기 정보의 추출을 통해 매칭을 위한 준비단계를 마치게 된다.

매칭 모듈의 경우 분석된 조각단위의 데이터의 유사도를 측정을 위한 5가지 매칭 기준을 사용하여 음악에 최적화된 영상 조각을 선택하게 된다. 만약, 적당한 매칭을 찾지 못하였을 경우 음악의 다중 분할 기법을 사용하여 새로운 매칭을 찾을 수 있다. 전체적인 시스템 개요는 그림 1에 나와있다.

4. 영상분할 및 분석

영상은 정보 전달을 위한 움직이는 이미지의 집합이다. 만약 음악에 맞는 비디오의 제작을 위해 비디오 클립의 임의의 부분을 단순히 붙여가는 과정만 계속한다면 촬영자가 원하고자 했던 정보전달의 의미가 깨질 위험성이 있다. 따라서 우리는 비디오의 흐름을 최대한 유지하기 위해 영상을 조각화한 뒤 분석하겠다.

4.1 외형매칭 기반의 영상분할

프레임간의 유사성이란 두개의 이미지가 얼마나 비슷한 색을 가지고 있느냐로 구분된다. 하지만 움직이고 있는 영상에서는 단순히 같은 좌표의 색상차이만으로는 유사성을 따지기 힘들다. 따라서 우리는 외형매칭(Contour shape matching) [5] 을 통한 유사성 측정을 통해 비디오를 분할한다. 임의의 N개의 이미지로 이루어진 영상조각 $V_i(i = 1, ..., N)$이 주어졌을때, 우리는 캐니 윤곽선 추출 기법(Canny edge detector) [6] 을 통해 이미지를 윤곽선 이미지 F_i로 변환 시켰다. 노이즈로 부터 발생할 수 있는 작은 외곽선을 방지하기 위해 영상클립을 미리 가우시안 필터링 해주었다. F_i는 픽셀 단위의 외곽선들로 이루어져 있다. 외곽선으로 부터 얻어진 7개의 Hu-moment를 $h_u^i(u = 1, ..., 7)$라 했을때 두 이미지의 외형차이는 다음과 같이 나타낼 수 있다.

$$I_{i,j} = \sum_{k=1}^{7} |1/m_k^i - 1/m_k^j| \quad (1)$$
$$where$$
$$m_k^i = sign(h_u^i) \log_{10} |h_u^i|$$

여기서, Hu-moment는 이동, 회전, 그리고 크기에 독립적이기 때문에 움직이는 영상에서 흐름이 끊기지 않는다면 유사

하다고 판정을 한다 [5]. 위의 식에서 얻어진 유사행렬 $I_{i,j}$는 그림 2 (a)에 나타나 있다.

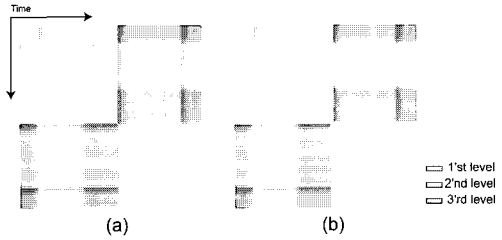

그림 2: 비디오 분할: (a) 유사행렬 $I_{i,j}$. (b) Radial symmetric kernel의 크기 조절을 통한 다중분할 결과 (δ=128,64,32).

Foote [1] 등은 그림 3 와 같은 radial symmetric kernel(RSK)를 통해 유사행렬을 분할하는 기법을 소개하였다. 우리는 앞에서 얻어진 유사행렬 $I_{i,j}$를 이용하여 대각선 방향으로 RSK를 적용하게 되면, 프레임 i에서의 앞뒤 프레임의 흐름의 차이값이 다음과 같이 구해진다.

$$E(i) = \sum_{u=-\delta}^{\delta} \sum_{v=-\delta}^{\delta} RSK(u,v) \cdot I_{i+u,i+v} \qquad (2)$$

여기서 δ는 커널의 크기를 뜯한다. 얻어진 $E(i)$값에서 정해진 임계치값 이상이 되는 극점을 찾아 분할의 기준으로 삼는다. 만약 δ값의 크기를 변화시킨다면 분할 정도를 조절할 수 있다. 큰 δ값을 쓸수록 짧은 변화는 무시가 되어 긴 길이를 가진 영상 조각을 생성할 수있다. 6 절에서 설명할 다중 레벨 매칭을 위해 우리는 δ값과 threshold값을 3단계로 나누어 적용한 후, 결과를 저장하였다 (그림 2 (b) 참조).

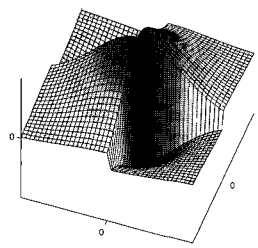

그림 3: Radial symmetric gaussian kernel(RSK)의 형태

4.2 영상의 특성 분석

영상은 촬영 기술 또는 압축정도에 따라서 많은 노이즈를 가지기 때문에 단순한 픽셀의 색 비교로는 정확한 움직임 분석이 힘들다. 따라서 본 논문에선 이미지의 윤곽선 기반의 비디오 움직임 측정기법을 제시한다. 윤곽선 기반의 움직임 측정을 위해 윤곽선 위의 한 점에 대하여 w의 크기를 가지는 윈도우 $\phi_{x,y}(p,q)$ (p,q는 윈도우 내에서의 좌표)를 선언해 준 후, 다음과 같은 색차이 추출식을 통해 움직임 벡터를 추출한다.

$$D^2 = \left(\phi^i_{x,y}(p,q) - \phi^{i+1}_{(x,y)+vec^i_{x,y}}(p,q)\right)^2 \qquad (3)$$

여기서, i번째 프레임과 $i+1$번째 프레임의 컬러의 차이인 D^2를 최소화 하는 벡터 $vec^i_{x,y}$가 윤곽선위의 한 점 (x,y)에 대한 움직임 벡터가 된다. 윤곽선이 아닌 부분의 움직임은 무시하기 위해 $F_i(x,y) = 0$인 pixel에서는 벡터를 $(0,0)$으로 고정해 준다. 보다 안정적인 결과를 위하여 지역적인 Lucas-Kanade [7] 기법을 적용하여 결과를 보완하였다.

윤곽선 이미지 F_i 를 사용한 속도 측정의 장점은, 윤곽선 위의 점끼리만의 비교이기때문에, 측정에 있어서 노이즈의 영향을 적게 받는다는 점이다. 하지만, 시청자가 쉽게 집중하지 않는 배경에서도 윤곽선이 생성될 수 있고 이것이 속도 측정에 영향을 줌으로써 분석의 정확성을 떨어뜨릴 위험성이 있다. 따라서 우리는 집중도 맵(saliency map)기반의 이미지 집중점 탐색 기법을 통해 주요 부분의 움직임 벡터를 강조하려 한다.

영상의 집중도 맵을 구성하기 위해 우리는 두가지 측면을 고려하였다. 첫째는 하나의 이미지 상의 이웃점과의 차이를 고려한 공간적 집중도이고, 다른 하나는 프레임간의 집중도를 고려하는 시간적 집중도이다. 우선 공간적 집중도는 Itti [8] 가 제안한 가우시안 거리(Gaussian distance)를 기반으로 계산하였다. 가우시안 거리의 공식은 다음과 같이 나타 낼 수있다.

$$G^i_s(x,y) = G_u(x,y) - G_{u+\eta}(x,y) \qquad (4)$$

여기서 G_u는 가우시안 피라미드상의 u번째 단계를 뜻한다. 즉 낮은 주기 데이터와 높은 주기 데이터의 차이를 통해 상대적으로 변이가 많은 부분을 찾을수 있다. 집중 정도는 밝기에 가장 영향을 많이 받으므로 [9] 각각의 프레임 V_i의 색상을 YUV공간으로로 변환 한 후 위의 식에 대입하였다.

시간적 집중도는 특정 위치에서의 픽셀의 움직임이 앞뒤 프레임과 얼마나 차이가 나는지의 고려를 통해 유추할 수 있다. 우리는 앞에서 구한 움직임 벡터 $vec^i_{x,y}$를 통해 움직임의 차이를 다음과 같이 구해내었다.

$$T^i_s(x,y) = N(||vec^{i-1}_{x,y} - vec^i_{x,y}||) \qquad (5)$$

여기서 N은 데이터를 정규화를 뜻한다. 위의 식은 i번째 프레임에서 (x,y)위치의 pixel이 가지는 가속도를 뜻한다. 가속도의 크기가 클수록 i번째 프레임의 집중성이 강해진다고 가정하였다. 여기서 한가지 더 고려해야 할 점은 카메라의 움직임 정보이다. 일반적인 촬영에 있어서 카메라를 움직인다는 것은 주요 물체를 추적한다는 뜻이 강하다. 정지된 카메라의 경우는 가속도가 클수록 그것이 집중도가 강한 부분이라고 볼 수 있지만, 카메라가 움직일 경우는 오히려 가속도가 작은 부분, 즉 카메라가 모션을 쫓아가는 부분이 더 큰 집중도를 가진다고 볼 수있다. 따라서 우리는 Lan 등이 [10] 제안한 ITM 기법을 통해 카메라 움직임 양을 측정하였다. 비디

(a) Input video (b) Edge detection (c) Vector Map (d) FOV region (e) Final Vector Map

그림 4: 윤곽선 기반의 비디오 속도 분석 : (a) 분할된 비디오 조각, (b) 윤곽선 추출 결과, (c) 움직임 벡터의 크기를 나타내는 맵, (d) 집중도 기반의 시야 정의, (e) 시야 외부의 움직임 벡터의 제거

오의 떨림을 고려하여 4픽셀이상의 카메라의 움직임이 있을 때 $T_s^i(x,y)$가 아닌 $1-T_s^i(x,y)$를 적용하였다.

얻어진 두개의 집중도 정도를 통해 우리는 사용자에게 주의를 줄수 있는 영역을 추출해 내려 한다. Ma [9] 가 제안한 방식을 이용하여 얻어진 집중도 정도를 기준으로 다음과 같은 식을 통해 i번째 프레임에서의 포커스 x_f^i, y_f^i를 찾을 수가 있다. 이미지를 $n \times m$크기라 가정하면:

$$x_f^i = \frac{1}{CM}\sum_{x=1}^{n} G_s^i(x,y)T_s^i(x,y)x \quad (6)$$

$$y_f^i = \frac{1}{CM}\sum_{y=1}^{m} G_s^i(x,y)T_s^i(x,y)y \quad (7)$$

$$where$$

$$CM = \sum_{x=1}^{n}\sum_{y=1}^{m} G_s^i(x,y)T_s^i(x,y)$$

우리는 fucus를 기준으로 전체 이미지 크기의 1/4 크기만큼의 시야를 정의하여, 안에 들어오는 vector들만 $\overline{vec}_{x,y}^i$에 저장하였다 (그림 4(d) 참조).

지금까지 구해온 tracking정보를 통해 우리는 프레임 i의 velocity값 V_{vel}^i를 다음과 같이 구할수있다. 전체 이미지에 대한 velocity는 다음과 같다.

$$V_{vel}^i = \frac{1}{n \times m}\sum_{x=1}^{n}\sum_{y=1}^{m} ||\overline{vec}_{x,y}^i|| \quad (8)$$

다음으로 영상의 밝기 정보를 추출하도록 하겠다. 이미지의 밝기정보는 프레임을 이루고있는 전체 픽셀의 밝기 분포를 통해 얻어내야 한다. 우리는 고전적으로 이용되고 있는 히스토그램 분석방식 [11] 을 사용하여 비디오의 밝기 정보를 추출하고자 한다. 우선 V_i를 흑백레벨로 변환한 뒤 밝기의 히스토그램을 분석해 낸다. 히스토그램을 10단계로 나누었을때 프레임을 대표하는 밝기값을 얻기위해 우리는 다음과 같은 식을 사용하였다.

$$V_{bri}^i = \sum_{u=1}^{10} B(u)^2 B_{mean_u} \quad (9)$$

여기서 $B(u)$는 히스토그램에서 u 단계에 담겨진 픽셀의 숫자이고 B_{mean_u}는 u 단계에 들어가 있는 픽셀밝기의 중간값이다. 실제 이미지에서 전체적으로 중간 밝기를 가지고있는 이미지보다 한쪽은 밝고 한쪽은 어두은 이미지가 더욱 밝은 느낌을 준다. 따라서 우리는 $B(u)$의 제곱항을 두어, 하나의 이미지에서 히스토그램의 단계의 차이가 크게 나면 날수록 더욱 큰 밝기 값을 가지게 하였다.

5. 음악분할 및 분석

분석된 영상과의 매칭을 위해선 음악 역시 조각단위로 잘라서 분석을 해야한다. 하지만 웨이브 형식에서 얻을수 있는 정보는 특정 시간 t에 대한 진폭 정보 밖에 존재하지 않는다. 따라서 주어진 진폭에 대한 시그널 분석을 통해 음악을 분할하고 속도와 밝기 정보를 측정해야한다.

음악의 분할을 위해 영상 분할과 유사한 방식으로 유사행렬을 구하도록 하겠다. Foote 등은 [1] 은 음악의 변화량을 구하기 위해 푸리에 변환을 사용한 novelty score라는 개념을 제안하였다. 주파수 영역에서의 유사도 측정을 통해 영상에서와 같이 RSK를 사용하여 급격한 변이점을 바탕으로 분할을 선택 할 수 있다. 그림 5 는 주어진 음악에 대한 주파수 영역에서의 유사도 분석 결과와, 구해진 Novelty score를 보여준다. 그림 5(b)에서 알 수 있듯이, RSK의 커널 크기를 조절함으로써 분할의 정도를 조절할 수 있다.

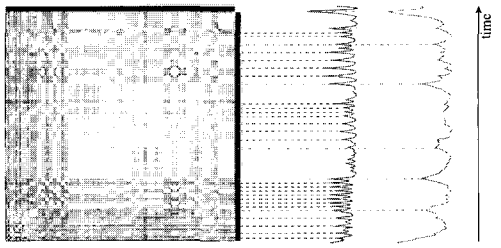

그림 5: 주파수 영역에서의 유사도 분석을 통한 novelty score생성: (a) 주파수 영영에서의 유사도 행렬 (b) 서로다른 크기의 커널을 가진 RSK을 사용하여 novelty score를 분석한 결과.

이제 얻어진 음악조각 $Mseg_k$에서 특징 정보를 추출해야 한다. Novelty score자체가 음악의 변이량을 나타내긴 하지만, 음악의 전환점에서만 큰 값을 가지므로 음악의 속도를

대변한다고 복긴 어렵다 (그림 6 참조). 따라서 우리는 웨이브의 진폭을 사용하여 속도를 분석하려 한다.

일반적으로 음악에서의 속도는 비트를 기반으로 한다. 기존에 많은 연구자들이 웨이브 신호를 바탕으로 비트를 찾는 연구를 진행되어왔지만 [12, 13], 우리가 필요로 하는것은 정확한 비트의 추적보다는 전체적인 음악의 빠르기를 목표로 한다. 따라서 신호의 진폭을 사용한 간단한 분석을 통해 음악의 빠르기를 유추하도록 하겠다. $Mseg_k$에서 시간 t에 대한 웨이브 시그널의 진폭값을 A_k^t라고 했을때, 음악을 비디오와 같은 샘플단위인 1/30초 단위의 윈도우로 분리한다. 분리된 조각에 대해 진폭의 중간값을 의미하는 root mean square(RMS) [14] 은 다음과 같이 계산된다.

$$RMS_i = \frac{1}{n}\sum_{u=1}^{n}(A_k^u)^2 \tag{10}$$

여기서 얻어진 RMS_i는 i번째 윈도우의 진폭값을 대표한다. 만약 악기가 빠른 속도로 연주가 될 경우에는 진폭값의 변이가 커지고, 느린속도로 연주가 될 경우에는 진폭값의 변이가 작아지게 된다. 따라서 음악의 속도를 유추하기 위해서 우리는 RMS의 변이량을 사용하여 다음과 같이 나타 낼 수 있다.

$$M_{vel}^i = |RMS_i - RMS_{i-1}| \tag{11}$$

이렇게 얻어진 음악의 빠르기 정보는 그림 6(b)에 나타나 있다. 그림 6(a)의 novelty score와는 달리 음악이 빠른 비트를 가지는 영역에서 RMS의 변이량은 커지고, 늦은 비트를 가지는 영역에서는 RMS의 변이량이 작아진다. 이런 현상의 원인은, 대부분의 음악이 타악기가 중심이 된 비트를 가지고 있고, 비트가 빠르면 빠를수록 타악기에서 생성되는 음향의 크기의 변이량이 커지기 때문이다.

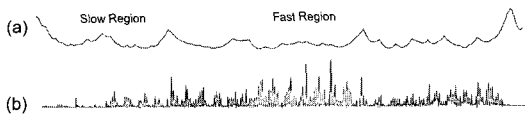

그림 6: Novelty score와 RMS의 변이량 : 음악이 갑자기 변하는 부분을 찾는데는 (a) novelty score가 좋은 기준이 되지만 음악의 빠르기를 추출하는 부분에선 (b) RMS의 변이량이 더욱 효과적이다.

다음으로 음악의 밝기 정보를 얻기위해 우리는 스펙트럴 중심(spectral centroid) [15] 정보를 사용하겠다. 스펙트럴 중심값은 일반적으로 사운드의 밝기를 측정하는데 사용되어진다. 음악에서의 밝기란 음색과 밀접한 연관성을 가지게 된다. 바이올린과 같이 고음을 내는 악기의 경우 밝은 소리를 내게 되고 베이스와 같이 저음을 내는 악기의 경우는 어두운 소리를 내게 된다. A_k^t 윈도우 단위로 쪼갠 뒤 주파수 영역에서의 분석을 통해, w를 주파수라 가정 했을때 아래와 같은 식을 통해아 음악의 밝기를 계산하였다.

$$A_{bri}^i = \frac{\sum w(A_i(w))^2}{\sum (A_i(w))^2} \tag{12}$$

6. 영상과 음악의 다중레벨 매칭

지금까지 영상과 음악의 매칭을 위한 분할과 각각의 조각의 속도정보와 와 밝기 정보를 추출하였다. 여기서 얻어낸 정보를 바탕으로 조각별로 최적의 매칭을 찾아내는 과정을 통해 뮤직비디오를 완성하려한다. 우리는 영상과 음악의 매칭을 위하여 얻어진 속도와 밝기정보를 기준으로 다음과 같은 5가지의 기준을 설정하였다.

- 급격한 변이점 동기화: 영상의 경우 Hu-momentum의 차이가 큰 부분에서 급격한 변이가 일어난다. 따라서 음악의 급격한 변이를 나타내는 novelty score와 매칭이 가능하다.
- 속도 동기화: 영상과 음악에 각각 구해진 V_{vel}^i값과 M_{vel}^i값의 유사성 측정.
- 밝기 동기화: 영상의 음악에 각각 구해진 V_{bri}^i값과 M_{bra}^i값의 유사성 측정.
- 분위기 동기화: 프레임내의 필셀들의 움직임을 나타내는 $\overline{vec}_{x,y}^i$ 값과 음악의 주파수 영역에서의 진폭값을 각각 히스토그램 분석을 통해 매칭.
- 시간 동기화: $Vseg_k$와 $Mseg_k$의 길이 값을 매칭

변이점, 속도, 밝기의 매칭의 경우 매칭의 기준이 되는 데이터가 분할된 조각의 크기에 의존적이기 때문에, 다른 시간 길이를 가진다. 따라서 각각의 데이터를 곡선을 통해 보간한 뒤 길이를 정규화 하여 비교하였다. 분위기 동기화의 경우, 우리는 기본적으로 작은 움직임이 많은 영상의 경우(예를들면 야외 풍경촬영물), 저음과 매칭이 되는것이 알맞다고 가정하였고 반대로 큰 움직임이 많은 복잡한 장면의 경우 고음과 매칭이 되는것이 알맞다고 가정하였다.

만약, k번째 음악조각 $Mseg_k$에 대해 알맞은 비디오 조각을 찾을 수 없을경우, 우리는 다중분할기법을 사용하여 $Mseg_k$를 다시 분할 한 뒤 새로운 매칭을 찾는다. $Mseg_k$의 재분할을 위해서 우리는 영상에서 사용한것과 같이 RSK의 커널 크기를 줄인 후 새롭게 얻어진 novelty score의 극점 사용하였다.

7. 실험결과

본 논문의 뮤직비디오 생성 시스템은 기본적으로 사용자에 의해 매칭을 위한 조건을 선택받게 된다. 앞절에서 설명한 5개의 매칭 조건을 사용하여 사용자가 각각에 대해 가중치를 할당함으로써 뮤직비디오의 결과를 조절 할 수 있다. 예를들어, 시간 동기화의 가중치를 줄이게 되면 서로다른 길이를 가지는 음악과 영상이 매칭될 확률이 커지게 되므로 역동적인 뮤직비디오가 만들어질 가능성이 커진다. 만약 급격한 변이점에 대한 가중치를 크게 한다면, 음악과 영상의 특징점에서의 매칭 확률이 상승하게 된다.

본 논문의 실험을 위해 우리는 총 30분 분량의 비디오를 촬영하여 100초 가량의 뮤직비디오의 자동생성을 시도하였다. 그림 7에서 나타나듯이 속도와 밝기면에서 다양성을 가지는 영상을 촬영하였고 음악도 변이가 큰 음악을 직접 작곡하여 실험을 해 보았다. 비디오는 총 142개의 조각들로 나누

어 졌고, 음악의 경우 최초에 11개의 조각들로 나누어 졌으나 다중레벨매칭에 의해 최종적으로 16개의 조각들로 나누어 져서 매칭이 이루어졌다.

표 1은 음악과 영상의 특징 분석을 위한 전처리 계산시간을 나타내고 있다. 음악의 경우는 1차원 시그널 분석이기때문에 오랜 시간이 걸리지 않지만 비디오의 경우 2차원 데이터이기 때문에 상대적으로 많은 시간이 걸리게 된다. 따라서 비디오 데이터의 경우 전처리로 추출된 속도와 밝기데이터를 미리 저장해 둠으로써 실제 뮤직비디오 생성 시간을 줄일 수 있다.

Media	Length	Segmentation	Velocity	Brightness
Video	30min	2.5hour	4.5hour	5min
Music	100sec	8.3sec	4.8sec	2.5sec

표 1: 영상과 음악의 전처리 계산 시간.

그림 7: 뮤직비디오 생성을 위한 예제 영상: 매칭의 정확도를 알기 위해 움직임과 밝기에 있어서 다양한 예제를 준비하였다.

8. 결론 및 향후계획

본 논문에선 영상의 연속성을 유지하기 위한 분할 기반의 자동 뮤직비디오 생성기법을 제안하고 있다. 동기화를 위해 영상 클립의 임의의 내용을 가져오는것이 아닌, 장면을 유지하는 조각 단위의 매칭을 통해 뮤직비디오의 연속성을 최대한 유지할 수 있다. 또한 다중분할매칭을 통해, 동기화의 정도를 더욱 상승 시킬 수 있었다. 본 논문의 자동 뮤직비디오 기법을 사용하여 비 전문적인 홈비디오 사용자도 간단한 매칭 가중치의 조정을 통해 손쉽게 뮤직비디오를 생성할수 있다.

본 논문의 한계점은, 매칭된 결과에 대해 동기화의 정도를 높일수 있는 도구를 제공하고 있지 않다는 점이다. 비록 가장 비슷한 매칭 정도를 가진다고 하여도, 우리의 시스템의 영상과 음악의 조각 전체의 속도와 밝기의 정보를 비교하지 않으므로, 정확한 타이밍에 동기화가 이루어 지지 않을 가능성이 있다.따라서 매칭된 조각끼리의 특징점 분석을 이용한 타임와핑 [16] 을 통해 동기화 정도를 높이는 방식을 도입하여 본 논문의 시스템을 더욱 발전시킬 수 있을 것이다.

조각단위의 영상과 음악의 동기화에 있어서, 하나의 조각에 들어있는 장면의 연속성은 보존할 수 있었지만, 전체적인 스토리를 보존할수는 없었다. 따라서 우리는 각 조각의 스토리적인 특성을 분석 또는 입력받아 최종적으로 뮤직 드라마를 만들수 있는 연구를 진행중이다. 또한 구조적인 영상이 입력으로 들어왔을 경우, 비디오 텍스쳐 [17] 를 사용하여 영상에 길이에 상관없이 동기화를 이룰 수 있는 뮤직비디오 생성 기법으로의 확장도 가능 할 것이다.

참고 문헌

[1] Foote J., Cooper M., and Girgensohn A. Creating music videos using automaticmedia analysis. In *Proceedings of ACM Multimedia*, pages 553–560, 2002.

[2] Zhang H. J. Hua X. S., Lu L. Ave - automated home video editing. In *Proceedings of ACM Multimedia*, pages 490–497, 2003.

[3] P. Mulhem, M. Kankanhalli, H. Hasan, and Y. Ji. Pivot vector space approach for audio-video mixing. *IEEE Multimedia*, pages 28–40, 2003.

[4] Jehan T., Lew M., and Vaucelle C. Cati dance: self-edited, self-synchronized music video. In *SIGGRAPH Conference Abstracts and Applications*, pages 27–31, 2003.

[5] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1963.

[6] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[7] Kanade T. Lucas, B. An iterative image registration technique with an application to stereo vision. In *Proceedings of 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[8] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1254–1259, 1998.

[9] Y.F. Ma and H.J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *Proceedings of 11th ACM international conference on Multimedia*, pages 374–381, 2003.

[10] H. J. Zhang D. J. Lan, Y. F. Ma. A novel motion-based. representation for video mining. In *Proceedings of IEEE. International Conference on Multimedia and Expo*, pages 6–9, 2003.

[11] Earl Gose, Richard Johnsonbaugh, and Steve Jost. *Pattern Recognition and Image analysis*. Prentice Hall, 1996.

[12] Masataka Goto. An audio-based real-time beat tracking system for music with or without drum-sounds. *Journal of New Music Research*, 30(2):159–171, 2001.

[13] Scheirer E.D. Tempo and beat analysis of acoustic musical signals. *Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[14] J. F. Kenney and E. S. Keeping. *Mathematics of Statistics, 3rd ed.* Princeton, 1962.

[15] Helmholtz H. L. *On the Sensation of Tone as a Physiological Basis for the Theory of Music (translation of original text 1877).* Dover Publications, 1954.

[16] H. C. Lee and I. K. Lee. Automatic synchronization of background music and motion in computer animation. In *Proceedings of the EUROGRAPHICS 2005*, pages 353–362, 2005.

[17] Arno Schodl, Richard Szeliski, David H. Salesin, , and Irfan Essa. Video textures. In *Proceedings of SIGGRAPH 2000*, pages 489–498, 2000.