

키넥트를 이용한 가상 바이올린 구현

Implementation of Virtual Violin with a Kinect

신영규*, 강동길*, 이정철*

Young-kyu Shin, Dong-gil Kang, Jung-chul Lee

요약

본 논문에서는 키넥트의 3차원 영상정보를 이용하여 종이 바이올린 지판과 손가락 끝점을 검출하고 활 움직임을 판정 한 뒤, 이 정보를 이용하여 가상 바이올린을 구현하는 방법을 제안한다. 키넥트의 컬러영상과 깊이영상을 이용하여 먼 저 바이올린 지판을 식별하고 손가락 끝점을 검출한 뒤 지판 정보와 사용자의 지판 누름 여부를 판정하기 위한 지판의 깊이 정보를 검출한다. 운지와 활 움직임 정보에서 음이름을 판정하여 PC MIDI 인터페이스를 통해 가상 바이올린 연주 시스템을 구현하였다.

본 논문에서 제안하는 방법을 이용하여 가상 바이올린 성능평가를 수행한 결과 높은 검출 정확도를 보였다. 구현된 가상 바이올린 기능을 활용하여 바이올린 연주 입력장치를 구현함으로써 PC기반 바이올린 연주시스템 구현의 편리성을 보였으며 악기 초보자가 PC 기반 바이올린 연주 학습에 사용자 인터페이스로 활용할 수 있는 가능성을 확인하였다.

ABSTRACT

In this paper, we propose a virtual violin implementation using the detection of bowing and finger dropping position from the estimated finger tip and finger board information with the 3D image data from a Kinect. Violin finger board pattern and depth information are extracted from the color image and depth image to detect the touch event on the violin finger board and to identify the touched position. Final decision of activated musical alphabet is carried out with the finger drop position and bowing information. Our virtual violin uses PC MIDI to output synthesized violin sound. The experimental results showed that the proposed method can detect finger drop position and bowing detection with high accuracy. Virtual violin can be utilized for the easy and convenient interface for a beginner to learn playing violin with the PC-based learning software.

Keywords : String Instrument, Kinect, Image Processing, Virtual Reality

1. 서론

바이올린의 경우 피아노와 달리 일상에서 경험하기가 쉽지 않고 배우기도 쉽지가 않다. 이러한 문제를 해결하기 위해 가상현실과 악기 연주를 접목시키는 소프트웨어들이 출시되고 있다[1-5]. 그러나 이들 소프트웨어는 PC 혹은 휴대형 단말기에서 출력되는 동영상, 혹은 GUI를 이용하여 바이올린 연주 학습법을 익히는 도구로 활용 가능하지만 학습자가 실제 바이올린 연주를 위한 손동작을 학습하기는 부족하다.

따라서 PC기반 바이올린 연주 학습프로그램이 종이에

그려진 가상의 바이올린을 연주하는 동작을 인식할 수 있다면 바이올린 학습에 유용하게 사용할 수 있을 것이다. 이를 위하여 영상정보를 이용하여 종이에 인쇄된 바이올린 지판을 인식하고 연주자의 손 영역을 검출하여 운지와 활 움직임을 판정할 수 있는 연구가 필요하다.

손 영역 검출과 가상악기 연주에 활용을 위한 연구로는 이차원 영상을 이용하는 방법과 삼차원 영상을 이용하는 방법이 있다. 이차원 영상을 이용하는 방법으로는 카메라의 영상데이터로부터 피부색을 고려하여 손가락을 인식한다[6,7]. 그러나 이 방법은 손가락이 key를 누르고 있는지 아닌지를 판단할 수 없다는 단점이 있다.

영상데이터를 처리하여 가상 바이올린 연주를 구현하기 위해서는 바이올린 운지부분과 활을 움직이는 손을 검출해야 한다. 그러나 2차원 영상데이터로 이러한 정보를 검출하는 것은 오류발생률이 매우 높아 3차원 영상데이터의 사용이 요구된다. 이차원 영상에서 다양한 배경화면으로부터 손 영역을 추정하는 어려움을 극복하기 위하여 3차원 입체 영상을 이용한 연구가 진행되었다[8-10].

* 울산대학교

본 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업지원금을 받아 수행되었음(과제번호: 2011-0015125).

투고 일자 : 2014. 7. 4 수정완료일자 : 2014. 7. 27

게재확정일자 : 2014. 8. 2

연구용 삼차원 영상데이터 수집장비는 고가이므로 본 연구에서는 비교적 저렴한 삼차원 영상데이터 수집장비인 키넥트를 사용하여 가상 바이올린을 구현하는 방법을 제안한다. 키넥트는 Microsoft사가 Windows용 SDK를 공개하면서 의학 분야에서는 뇌졸중 재활 치료 운동 프로그램, 교육 분야에서는 모션 인식 기능을 이용한 수화번역 시스템 등 다양한 분야에 사용되고 있다[11-13]. 그러나 아직 키넥트를 이용한 바이올린 연주 프로그램은 현재 상용화된 것이 없다.

본 연구에서는 Kinect for Windows SDK를 이용하여 C# 기반의 가상 바이올린 연주를 개발하였다. 종이에 인쇄된 바이올린 모형을 가상으로 연주하는 삼차원 영상데이터를 키넥트를 이용하여 수집한 뒤 영상처리를 이용하여 바이올린 운지부분과 활을 움직이는 손을 검출하고 연주정보를 추정한 뒤, PC MIDI를 이용하여 바이올린 음을 출력하도록 구현하였다. 키넥트의 깊이 데이터만을 이용하여 손 영역과 음 포지션을 검출할 경우 정확도 떨어지므로 정확도를 향상시키기 위한 새로운 검출 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. II장에서는 전체적인 가상 바이올린 시스템을 소개하고 세부 모듈에 대한 설명과 구현방법에 대해서 자세히 설명한다. III장은 실험 및 결과에 대하여 설명하고 마지막으로 IV장에서 결론과 향후 과제에 대해서 요약한다.

II. 본 론

본 논문에서 제안한 가상 바이올린은 그림 1과 같이 제어모듈, 키넥트 프로그래밍 모듈, 영상 처리 모듈, MIDI 모듈, 화면 출력 모듈로 구성된다.

메인 어플리케이션은 C# 클래스로 만들어진 윈도우 생성 및 각종 이벤트 처리를 담당한다. 키넥트 프로그래밍 모듈은 키넥트 초기화 과정, calibration 과정을 수행하고 컬러영상, 깊이영상, 관절위치 데이터들을 초당 30프레임 생성하여 PC로 전송한다. 영상처리모듈에서는 키넥트로부터 바이올린 연주자의 연주동작과 관련된 삼차원 영상 데이터를 전달받아 가상 바이올린 연주를 인식하는데 필요한 데이터를 검출하고 PC MIDI 장치로 메시지를 출력하여 바이올린 음을 생성한다. 그리고 PC 화면에 그려진 바이올린의 운지점에 붉은색으로 표시하여 사용자가 짚은 음을 확인할 수 있게 구현하였다.

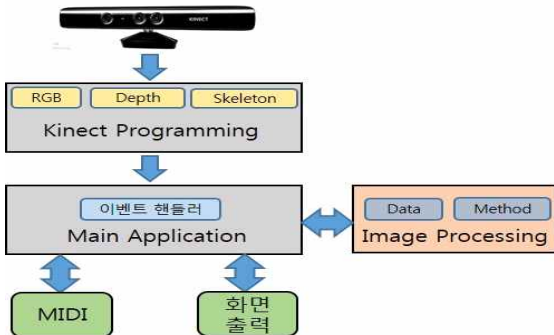


그림 1. 제안하는 가상 바이올린 구현 방법
Fig. 1. Proposed virtual violin

2.1 키넥트 프로그래밍 모듈

본 연구에서는 키넥트 센서를 제어하기 위해서 마이크로소프트사의 Kinect for Windows SDK에서 제공되는 라이브러리를 이용하였다.

영상처리모듈에서 바이올린 운지와 활의 움직임 검출을 위해서 사용되는 컬러영상, 깊이영상, 관절위치 정보를 생성하기 위해서 먼저 키넥트 초기화 과정에서 ColorStream, DepthStream, SkeletonStream을 모두 활성화한다. 그리고 Near 모드를 선택하여 최소 40cm 근거리에서 깊이영상 획득이 가능하도록 하였으며, 의자에 앉아 바이올린 연주가 가능하도록 skeleton 추적 모드를 seated 모드로 설정하였다. 또한 검출된 관절위치가 프레임 간에 급격히 변화하지 않도록 smoothing 기능을 사용하였다. 영상 프레임 크기는 640x480 픽셀, 초당 30 프레임으로 설정하였다. 키넥트의 컬러 카메라와 깊이영상 카메라 간 거리 차로 인한 깊이영상과 컬러영상의 좌표 불일치가 발생하는데 이러한 불일치를 MapDepthFrameToColorFrame 메소드를 사용하여 보정하였다.

키넥트 센서를 성공적으로 초기화하게 되면 키넥트 센서는 초당 30번 컬러영상, 깊이영상과 스켈레톤 데이터를 생성하며 메인 어플리케이션은 키넥트의 이벤트 핸들러를 이용하여 생성된 데이터를 전달받아 영상처리 과정에 사용한다.

2.2 영상처리 모듈

영상처리 모듈에서는 컬러영상, 깊이영상과 스켈레톤 데이터를 분석하여 손 영역을 검출하고 바이올린 운지와 활 움직임을 인식한 결과를 메인 어플리케이션으로 보낸다.

(a) 바이올린 운지영역 검출

바이올린 운지 영역을 검출하기 위해 1단계에서 키넥트의 관절위치 데이터 중에서 연주자의 head 위치정보를 기준으로 왼쪽의 일정 영역을 바이올린 운지 후보 영역으로 설정하였다. 운지 후보 영역은 바이올린의 길이와 사람의 신체 크기를 고려하여 그림 2(a)와 같이 사각형 영역으로 설정하였다. 그리고 2단계에서는 그림 2(b)와 같이 후보 영역 내의 영상에서 깊이영상을 이용하여 키넥트와 연주자 head 간의 거리를 기준으로 일정한 거리 영역 밖에 있는 영상 정보를 제거함으로써 손과 가상 바이올린 영상만 남게 하였다. 남겨진 영상을 이용하여 다음 단계에서 손 영역과 바이올린 음 포지션을 검출하게 된다.

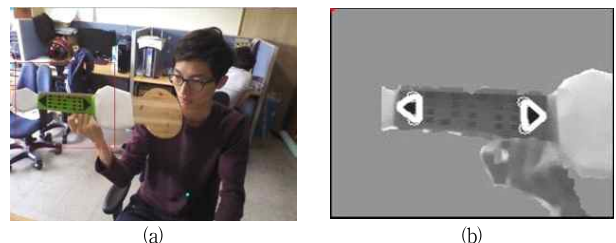


그림 2. 바이올린 지판과 손 영역 검출 (a) 1단계, (b) 2단계
Fig. 2. Finger board and hand detection (a) stage 1, (b) stage2

(b) 손가락 끝점 검출

바이올린 운지를 검출하기 위해서는 연주자의 손가락 끝점을 검출하여야 한다. 키넥트는 손가락 관절의 위치를 검출하지 못하므로 다음과 같은 방법을 적용하였다.

1단계는 키넥트의 컬러영상으로부터 손 영역을 검출한다. 바이올린의 거리 정보를 기준으로 이용하여 손영역을 구분할 수도 있지만 실제 연주시 바이올린의 위치가 시간에 따라 변하게 되므로 적용하기 어렵다. 본 연구에서는 RGB 데이터를 빛의 밝기에 덜 민감한 YCbCr 컬러 데이터로 변환한 후 피부색 영역값을 이용하여 그림 3과 같이 손영역을 검출하였다. 피부색 영역 값은 실제 컬러영상 450장에서 수작업으로 분리한 손 영역의 Cb, Cr 값을 통계 처리하여 설정하였다. 2단계에서는 1단계에서 검출된 손영역으로부터 손가락 끝점을 검출한다. 이를 위해서 피부색 영역에서 레이블링 알고리즘을 이용하여 잡티 현상을 제거함으로써 손영역을 검출하였고 모폴로지 알고리즘의 Closing 연산 처리 방법으로 손영역을 평활화 하였다. 그리고 ConvexHull 함수를 사용해 손 영역의 외곽선을 검출한 후 손가락 끝점을 그림 3과 같이 검출하였으며 그림에서 검출된 손가락 끝점을 둥근 점으로 표시하였다. 영상 처리 과정에는 Opencv 라이브러리를 사용하였다[14,15].



그림 3. 영상 처리 후 손가락 끝점 검출

Fig. 3. Finger tip detection result after image processing

(c) 바이올린 음 포지션 검출

바이올린 음 포지션 검출은 그림 4와 같이 바이올린 모형에 표시된 양쪽 검은색 두 삼각형을 기준으로 이루어진다. 실제 바이올린 음 포지션들은 일정한 길이로 배치되어 있고 이를 이용하여 4개 좌표 기준으로 가상 바이올린 음 포지션을 만들었다.

바이올린 모형의 두 검은색 삼각형을 감지하기 위해 RGB 데이터를 그림 4(a)와 같이 gray 스케일로 변경한다. 검은색 임계치를 이용하여 영상을 2진화하고 레이블링 알고리즘으로 두개의 삼각형 영역을 검출한다. 모폴로지 알고리즘인 Closing 연산을 통해 부드럽게 만든 후 외곽선을 검출함으로써 삼각형 끝점들의 좌표 값을 구할 수 있다. 검출된 삼각형 좌표 값을 이용하여 바이올린 지판의 영역을 설정하였고 이를 기준으로 균등하게 분할함으로써 그림 4(b)와 같이 바이올린의 음 포지션 영역을 설정하였다. 실제 바이올린을 연주할 경우 바이올린이 고정되지 않고 움직이므로 매 프레임마다 포지션 영역을 검출하였다.

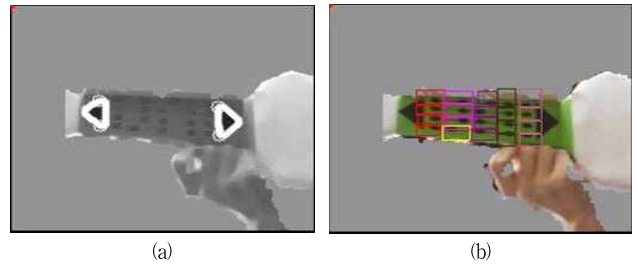


그림 4. 바이올린 음 포지션 검출 (a) 지판 영역 설정, (b) 음 포지션 검출

Fig. 4. Violin pitch position detection (a) finger board detection, (b) pitch position detection

(d) 연주음 판정

검출된 손가락 끝점 좌표와 바이올린 음 포지션 패턴 정보를 이용하여 손가락 끝이 위치하는 음의 위치를 식별할 수 있다. 그러나 연주자가 손가락으로 누름 여부를 판별하기 위해서는 추가 정보가 필요하다. 본 연구에서는 손가락 끝점의 깊이 값과 바이올린 음 포지션 영역의 깊이 정보를 비교하여 절대오차가 지판의 위치를 검출함으로써 바이올린 연주음을 인식하는 알고리즘은 키넥트의 깊이 영상 값을 이용한다. 각 포지션들은 깊이 값을 가지며 이 깊이 값과 손가락 끝점 깊이 값의 차이가 5mm 이하 일 때 눌러 졌다고 판정을 한다.

그러나 손가락 끝점과 음 포지션 검출만으로 운지를 하게 되면 부정확한 판정이 빈번히 발생하였다. 이러한 문제를 해결하기 위해 본 논문에서는 아래와 같은 3가지 방법을 제안한다.

첫 번째 방법으로 정확한 바이올린 운지 판정을 위해 투표 방식을 사용하였다. 손가락 끝점 하나를 기준으로 삼지 않고 손가락 끝이라고 추정되는 인접한 점들을 이용하여 각 점에 대한 음 포지션을 구한 뒤, 가장 많은 투표를 받은 음 포지션이 눌러졌다고 판정하였다.

두 번째 방법은 음 포지션을 정비하는 알고리즘을 적용하였다. 바이올린 음 포지션은 양쪽 삼각형을 기준으로 포지션을 검출하기 때문에 삼각형의 깊이 값이 정확히 검출되지 않는 경우 오류가 발생한다. 이를 해결하기 위해 인접 프레임들의 양쪽 삼각형 X, Y 차이의 평균값을 계산하였다. 그리고 이를 기준으로 영상 처리 중 평균값을 기준으로 일정 범위를 벗어날 경우 이전 프레임의 포지션으로 대체함으로써 포지션 검출 오류를 방지하도록 구현하였다.

세 번째 방법으로는 손이 지판을 가려서 키넥트가 바이올린의 음 포지션 깊이 값을 검출하지 못하게 되어 판정이 어려운 경우에는 이전 프레임 깊이 값을 사용하여 판정하도록 하였다.

이상의 3가지 방법을 적용하여 손가락 끝이 포지션을 가렸을 경우나 키넥트가 운지를 한 음 포지션의 깊이 값을 놓쳤을 경우, 인접 프레임의 각 음 포지션 깊이 값을 사용함으로써 연주음 판정의 정확도를 향상시키도록 구현하였다.



그림 5. 활 움직임 (오른손 영역) 감지
Fig.5. Bowing detection (right hand)

(e) 바이올린 활 움직임 검출

바이올린 활의 움직임을 검출하기 위해서 관절위치 데이터 중에서 연주자의 오른손 관절의 위치정보를 사용할 수 있다. 그러나 손의 위치가 몸통과 겹쳐질 경우 손 관절의 추정에 오류가 빈번히 발생하여 신뢰도가 떨어지므로 사용하기 어려웠다. 이에 본 연구에서는 키넥트의 관절위치 데이터 중에서 연주자의 head 위치정보를 기준으로 그림 5와 같이 오른쪽의 일정 영역을 활 움직임 후보 영역으로 설정하였다. 활 움직임 후보 영역은 활의 길이와 사람의 신체 크기를 고려하여 설정하였다. 그리고 2단계에서는 후보 영역 내의 컬러영상과 깊이영상을 이용하여 손 영역을 검출하였다.

활 움직임을 검출하기 위해서 오른손의 움직임 값 M을 식 (1)과 같이 인접 연속 3 프레임의 오른손 끝점 좌표를 이용하여 구하였다. 실험을 통하여 움직임 감지의 임계값을 8로 설정하였다. 임계값을 기준으로 이보다 클 경우 활이 움직이는 상태로, 미만일 경우 멈춰 있는 상태로 판단하여 활의 움직임을 실제 바이올린의 연주처럼 보이도록 구현하였다.

$$Math.Abs(Bow[0].Y - Bow[1].Y) + Math.Abs(Bow[0].Y - Bow[2].Y) \quad (1)$$

2.3 출력 모듈

(a) MIDI 출력 모듈

MIDI 제어모듈에서는 가상 바이올린으로부터 연주음과 활의 움직임 검출 정보를 입력받아 해당되는 음표번호와 채널, 속도, 음표 on 정보로 MidiOutShortMsg를 생성하고 PC의 MIDI를 제어하여 바이올린 음을 출력한다[16]. 활 움직임이 정지되거나 이전에 눌러진 음이 선택해제되면 음표번호와 채널, 속도, 음표 off 정보로 MidiOutShortMsg를 생성하고 PC의 MIDI를 제어하여 바이올린 음의 출력을 중지한다.

(b) 화면 출력 모듈

본 연구에서 구현한 키넥트를 이용한 가상 바이올린 구현 결과는 그림 6과 같다. 키넥트로부터 받은 컬러영상을 왼쪽 상단에 그리고 사용자가 선택한 악보를 오른쪽 상단에 출력하여 사용자가 악보를 보면서 연주할 수 있게 하였다. 그리고 깊이영상과 컬러영상을 이용하여 검출된 연주음과 활의 움직임 정보를 이용하여 화면 하단에 그려진 바



그림 6. 키넥트를 이용한 가상 바이올린 구현결과 화면
Fig. 6. Result of virtual violin implementation with Kinect

이올린의 운지점에 붉은색으로 표시하고 오른쪽 하단 텍스트 창에 누른 음 이름을 출력한다. 매 프레임마다 영상과 검출한 결과를 화면에 갱신하였다.

III. 실험 및 결과

본 연구에서 제안한 내용을 바탕으로 프로그램을 구현하고 키넥트와 연동 실험을 통하여 성능 분석을 수행하였다. 연구개발 환경은 PC Windows7 환경에서 C# 프로그래밍 언어를 사용하여 프로그램을 개발하였다. 본 논문에서 제안한 방법의 성능을 평가하기 위해 초당 30 프레임으로 30초 간 영상데이터를 저장하여 수작업으로 손가락 끝점, 음 포지션, 운지점 누름, 활 움직임 정보를 작성하였다. 그리고 제안한 방법을 사용하여 검출된 데이터와 비교함으로써 성능을 판정하였다.

바이올린 운지를 감지하기 위해 가장 중요한 손가락 끝점을 검출한 결과는 표 1과 같이 76%의 정확도를 보였으며 음 포지션 검출은 72%의 정확도를 얻을 수 있었다.

손가락 끝 검출과 음 포지션 검출만으로 운지점을 검출했을 경우 실험결과는 표 2와 같다. 영상데이터 분석 결과 손가락이 음 포지션의 경계에 위치하는 경우, 손가락에 가려서 음 포지션에 관한 거리정보를 획득하지 못하는 경우,

표 1. 손가락 끝점, 음 포지션 검출 결과

Table 1. Detection results of finger tip and finger board

	실험 내용	정확도(%)
손가락 끝점 검출		76
음 포지션 검출		72

표 2. 손가락 끝점과 음 포지션을 이용한 운지점 검출 결과
Table 2. Detection results of finger dropping position from the estimated finger tip and finger board information

실제 누른 음	판정률(%)
하프포지션 과	77
2포지션 시	80
2포지션 솔	57
3포지션 솔	76
4포지션 레	81
4포지션 시	62
개방 현 레	88
개방 현 라	82
개방 현 미	53

그리고 음 포지션 마크가 손가락에 가려진 경우에 판정 오류가 발생하게 되어 평균 74%의 낮은 정확도를 보이고 있음을 알 수 있다.

본 논문에서 제안한 3가지 판정 알고리즘을 적용하였을 경우 실험결과는 표 3과 같다. 제안된 판정 알고리즘을 적용하였을 경우 평균 93%의 높은 정확도를 얻을 수 있었다. 판정 오류가 47%였던 ‘개방 현 미’의 경우 제안된 알고리즘을 적용한 뒤 오류가 13%로 낮아졌다. ‘개방 현 라’와 ‘레’의 경우 88%, 그 외 음들은 전부 90% 넘는 정확도를 보여 본 논문에서 제안한 알고리즘을 적용했을 경우 판정의 정확도가 크게 향상되는 것을 확인할 수 있었다. 그러나 음 포지션 마크에 가까운 개방 현 운지점을 누를 때 손가락에 의해 가려지면 오류 발생률이 여전히 높았다.

키넥트의 관절위치 데이터 중에서 오른손 관절위치 정보를 이용하여 바이올린 활의 움직임을 검출한 결과는 표 4와 같다. 활의 움직임 판정을 위해서 기준 영역을 정해놓은 후, 영역 안에 오른손 관절이 들어오게 되면 관절위치를 추적하여 손의 움직임을 판단하고, 영역 안에 가만히

표 3. 판정 알고리즘 적용 후 운지점 검출 결과
Table 3. Detection results of finger dropping position obtained by our proposed algorithm

실제 누른 음	판정률(%)
하프포지션 과	100
2포지션 시	100
2포지션 솔	100
3포지션 솔	94
4포지션 레	94
4포지션 시	90
개방 현 레	88
개방 현 라	88
개방 현 미	87

표 4. 키넥트 관절위치 정보를 이용한 활 움직임 검출 정확도
Table 4. Bowing detection results with skeleton information

활 움직임	정확도(%)
올라갈 때	71
멈춘 경우	54
내려갈 때	73

표 5. 제안된 알고리즘을 이용한 활 움직임 검출 정확도
Table 4. Bowing detection results with proposed method

활 움직임	정확도(%)
올라갈 때	97
멈춘 경우	88
내려갈 때	97

있게 되면 멈춘 상태로 판정하였다.

그러나 키넥트의 관절위치 정보를 이용하여 오른손을 추적할 경우 표 4와 같이 낮은 정확도를 나타내고 있다. 특히 멈춘 경우에는 54% 정도의 정확도를 보임으로써 정확한 활 움직임을 구현할 수 없음을 알 수 있었다.

본 논문에서 제안한 알고리즘을 적용할 경우 표 5와 같이 정확도가 향상된 결과를 보임으로써 활 움직임을 정확히 검출하는데 성공적임을 알 수 있었다.

VI. 결론

본 논문에서는 키넥트의 3차원 영상정보를 이용하여 종이 바이올린 지판과 손가락 끝점을 검출하고 활 움직임을 판정한 뒤, 이 정보를 이용하여 가상 바이올린을 구현하는 방법을 제안하였다. 키넥트의 컬러영상과 깊이영상을 이용하여 먼저 바이올린 지판을 식별하고 손가락 끝점을 검출한 뒤 지판 정보와 사용자의 지판 누름 여부를 판정하기 위한 지판의 깊이 정보를 검출한다. 운지와 활 움직임 정보에서 음이름을 판정하여 PC MIDI 인터페이스를 통해 가상 바이올린 연주시스템을 구현하였다.

본 논문에서 제안하는 방법을 이용하여 가상 바이올린을 구현하였으며 성능평가를 수행한 결과 높은 검출 정확도를 보였다. 구현된 가상 바이올린 기능을 활용하여 바이올린 연주 입력장치를 구현함으로써 PC기반 바이올린 연주시스템 구현의 편리성을 보였으며 악기 초보자가 PC 기반 바이올린 연주 학습에 사용자 인터페이스로 활용할 수 있는 가능성을 확인하였다. 미래 기술로 각광받는 삼차원 인체 인식 인터페이스를 잘 구현시킬 수 있는 키넥트의 장점을 활용하여 바이올린 연주 입력 장치를 구현함으로써 PC기반 가상 바이올린 연주시스템 구현의 편리성을 보였다.

향후 연구과제로는 운지 부분과 활 부분 움직임 인식을 개선하여 정확한 인식을 통해 부드러운 연주가 가능하게 할 것이다. 그리고 적용 악기의 범위를 확장하고 이를 이용한 게임 또는 초보자 학습 프로그램 개발에 적용할 예정이다. 또한 손동작 인식을 통한 가상현실 및 증강 현실 구현과 사용자 유저 인터페이스 개발에 활용하고자 한다.

참 고 문 헌

- [1] <http://takelessons.com/blog/apps-for-violinists>
- [2] <https://itunes.apple.com/us/app/learn-violin/id534046897?mt=8>
- [3] <http://www.violinlab.com/>
- [4] http://www.amazon.com/eMedia-EV12090-My-Violin/dp/B000VPRFSE/ref=sr_1_3?s=software&ie=UTF8&qid=1404390359&sr=1-3
- [5] http://www.doyac.com/new/01info/lect_list.php?lect_mode=violin
- [6] K.M. Cho, J.H. Jang, and K.S. Hong, "Adaptive Skin-Color Filter," Pattern Recognition, vol.34, no.5, pp.1067-1073, May 2001.
- [7] J.M. Jeong, J.R. Jang, Y.I. Kim, J.W. Park, and W.J. Lee, "Development of the Hand Recognition System for the Mouse Control," Proceeding of Winter Conference of KSCI, vol.19, no.1, pp.173-174, Jan. 2011.
- [8] S.W. Jang, Y.J. Park, and G.Y. Kim, "Human Skin Region Detection Utilizing Depth Information," Journal of The Korea Society of Computer and Information, vol.17, no.6, pp.29-36, June 2012.
- [9] S.Y. Cho, H.R. Byun, H.K. Lee, and J.H. Cha, "Hand Gesture Recognition from Kinect Sensor Data," Journal of Broadcast Engineering, vol.17, no.3, pp.447-458, May 2012.
- [10] J.C. Lee and M.S. Kim, "Implementation of Paper Keyboard Piano with a Kinect," Journal of The Korea Society of Computer and Information, vol.17, no.12, pp.221-230, December 2012.
- [11] D. Catushe, Programming with the Kinect for Windows Software Development Kit, Microsoft Press, 2012.
- [12] J. Webb and J. Ashley, Beginning Kinect Programming with the Microsoft Kinect SDK, Apress, 2012.
- [13] J. Hall, S. Kean, and P. Perry, Meet the Kinect: An Introduction to Programming Natural User Interfaces, Apress, 2011.
- [14] G. Bradski, Learning Opencv: Computer Vision with the Opencv Library, O' REILLY, 2008.
- [15] R.C. Gonzalez, Digital Image Processing, 3rd edition, Pearson, 2009.
- [16] P. Messick, Maximum MIDI: Music Applications in C++, Manning Publications, 1998.



신 영 규 (Young-kyu Shin)

2009년 3월 ~ 현재 울산대학교
전기공학부(학사과정)
※주관심분야 : 디지털신호처리



강 동 길 (Dong-gil Kang)

2009년 3월 ~ 현재 울산대학교
전기공학부(학사과정)
※주관심분야 : 디지털신호처리



이 정 철 (Jung-chul Lee)

正會員

1984년 2월 서울대학교 전자공학과
(공학사)
1988년 2월 서울대학교 전자공학과
(공학석사)
1998년 8월 서울대학교 전자공학과
(공학박사)

1985년~2000년: ETRI 책임연구원
2000년 : L&H Korea 전문위원
2001년 : (주)보이스텍 전문위원
2002년 : (주)코난테크놀로지 책임연구원
2002년~현재: 울산대학교 전기공학부 부교수
※주관심분야 : 신호처리, 음성코딩, 음성합성