

키넥트를 이용한 종이건반 피아노 구현 연구

이정철*, 김민성*

Implementation of Paper Keyboard Piano with a Kinect

Jung-Chul Lee *, Min-Seong Kim *

요 약

본 논문에서는 키넥트의 3차원 영상정보를 이용하여 손가락 움직임을 검출하고 이 정보를 이용하여 종이건반 피아노를 구현하는 방법을 제안한다. 키넥트의 컬러영상과 깊이영상을 이용하여 먼저 건반 식별을 위해 필요한 건반 패턴 정보와 사용자의 종이건반 누름 여부를 판정하기 위한 건반깊이 정보를 검출한다. 배경의 깊이정보를 이용하여 입력된 깊이정보로부터 손 영역을 검출할 경우 종이건반과 접촉한 손가락 끝부분이 잘려나가는 문제를 해결하기 위해 스킨 컬러를 이용하여 정확도를 향상시켰다. 그리고 면적을 이용한 외곽선 검출과convex hull 알고리즘을 이용하여 손가락 끝점을 검출하고 건반 패턴 정보와 비교하여 건반 누름을 판정하였다.

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

▶ Keywords : 건반악기, 키넥트, 영상처리

Abstract

In this paper, we propose a paper keyboard piano implementation using the finger movement detection with the 3D image data from a kinect. Keyboard pattern and keyboard depth information are extracted from the color image and depth image to detect the touch event on the paper keyboard and to identify the touched key. Hand region detection error is unavoidable when using the simple comparison method between input depth image and background depth image, and this error is critical in key touch detection. Skin color is used to minimize the error. And finger tips are detected using contour detection with area limit and convex hull. Finally decision of key touch is carried out with the keyboard pattern information at the finger tip position.

• 제1저자 : 이정철 • 교신저자 : 이정철
• 투고일 : 투고일 : 2012. 12. 5, 심사일 : 2012. 12. 17, 게재확정일 : 2012. 12. 20.
* 울산대학교 전기공학부(School of Electrical Engineering, University of Ulsan)

The experimental results showed that the proposed method can detect key touch with high accuracy. Paper keyboard piano can be utilized for the easy and convenient interface for the beginner to learn playing piano with the PC-based learning software.

▶ Keywords : Keyboard Instrument, Kinect, Image Processing

I. 서론

음악학습에서 컴퓨터기반 전자악기를 이용할 경우 자율학습이 가능하여 학습의 효율을 향상시킬 수 있다. 그러나 상용화된 전자악기와 PC MIDI를 이용한 소프트웨어들은 상호 독립적이기 때문에 서로 연동이 되지 않을 뿐만 아니라 고가이며 해당 제품을 사용함에 있어 전문성이 요구되어 초등학교 저학년 학생들이 악기 연주 및 학습에 어려움이 있다[1-4]. PC를 기반으로 한 피아노 연주 프로그램의 입력장치로는 키보드, 마우스가 있지만 사용하는데 불편하고 실제 피아노 건반 배열과는 큰 차이가 있다. 마스터 키보드와 같은 외장 전자 건반장치를 이용하여 입력할 경우는 실제 피아노와 거의 동일한 입력이 가능하지만, 피아노 연주를 위해 특별히 만들어진 하드웨어로서 가격이 비싸고, 공간을 많이 차지하며 기능의 추가 등이 제한된다. 프로젝션 키보드의 경우 작은 소형 디바이스에서 가상의 키보드를 평면에 투영하고 적외선 센서를 이용하여 인식하는 방식으로 키보드 인식에 특화되어 있다.

따라서 PC기반 피아노연주 학습프로그램이 종이에 인쇄된 피아노 건반을 학습자가 누르는 동작을 인식할 수 있다면 사용자 입력의 편리성을 높일 수 있을 것이다. 이를 위해 영상 정보를 이용하여 종이에 인쇄된 피아노 건반을 인식하고 연주자의 손 영역을 검출하여 건반 누름을 판정할 수 있는 연구가 필요하다.

손 영역 획득을 위한 기존 연구방식에는 데이터 글로브 센서를 이용하는 방식과 카메라 영상을 이용하는 방식이 있다. 데이터 글로브는 착용으로 인해 불편과 고가인 단점들이 있다. 카메라 영상은 단일 카메라를 이용하는 2D, 복수 카메라를 이용하는 3D로 구별된다.

2D 영상의 경우 일반적으로 카메라의 RGB 영상을 HSI 혹은 YCbCr 컬러로 변환한 뒤 피부색상의 범위에 있는 영역을 손 영역으로 판정하는 단순한 방식이지만 유사 색상에 의한 오인식율이 높다[5]. 이에 손 모양으로부터 확장된 하얼 특징을 추출하여 Adaboost 알고리즘으로 손 영역을 탐지하고 칼만 필터를 이용하여 탐색 영역을 예측하는 방법이 제안

되었다[6,7]. 그러나 장시간의 학습이 필요하고 물체가 접촉 경우 오류가 발생하는 문제가 있다. 배경영상과 입력영상의 차영상을 이용하여 움직임 영역을 구하고, HSV 컬러모델을 이용하여 피부색상과 유사한 영역을 구한 뒤, 교집합을 통하여 손 영상을 추출하는 방법이 제안되었다 [8]. 그러나 이 방법도 배경화면의 변화에 지속적으로 영향을 받는 단점이 있다.

2D 영상에서 다양한 배경화면으로부터 손 영역을 추정하는 어려움을 극복하기 위하여 3차원 입체 영상을 이용한 연구가 진행되었다. 스테레오 영상으로부터 추정된 깊이정보와 색상 기반의 2차원적인 피부영역 정보를 결합하여 3차원의 피부영역을 검출하는 방법이 제안되었다[9]. 그러나 깊이정보의 정확한 추정이 어려운 문제가 있다.

최근 마이크로소프트사에서 게임용 동작인식을 위한 NUI (Natural User Interface)로서 저가의 키넥트 (Kinect) 센서를 출시하였다. 키넥트 센서는 RGB 카메라, 적외선 프로젝터와 연동되는 모노크롬 CMO 센서를 이용한 깊이 측정 센서를 장착하고 있어 최근의 3D 카메라 기능을 이용한 연구에도 활용되고 있다[10]. 키넥트 센서로부터 획득한 어깨, 팔꿈치 및 손 등의 관절 정보를 이용하여 다각도 결합 히스토그램을 구성한 뒤 랜덤 결정 포레스트 분류기의 학습하여 손 제스처를 인식하는 방법이 제안되었다[11]. 그러나 대부분의 몸체가 키넥트 센서의 시야 내에 존재해야 관절에 대한 위치 정보를 제공받을 수 있으므로 paper 피아노 구현에는 적합하지 않다. 다른 접근 방법으로는 키넥트 3D 센서의 색상 및 깊이 정보를 융합한 Mean-Shift 추적 기법을 사용하여 복잡한 배경에서 손을 검출하는 방법이 제안되었다[12,13]. 이 연구는 비마커 증강현실을 위한 손 자세 추정을 목표로 하고 있어 종이건반 누름 판정에는 적합하지 않다.

키넥트를 이용한 가상피아노 구현과 관련된 연구결과는 youtube상에서 볼 수 있는데 대부분 허공에 가상건반이 있음을 가정하고 손가락 움직임을 감지하여 구현하고 있어 저학년 피아노 연주학습에 적용하기 어렵다[14].

본 논문에서는 종이에 그려져 있는 피아노 건반 그림을 인식하고 사용자 손가락이 건반을 누르는 것을 인지하여 PC MIDI를 구동하는 종이 피아노 구현 방법을 제안한다. 종이

피아노 구현을 위해서는 최근에 동작인식 디바이스로서 각광 받고 있는 키넥트를 이용하며 근거리에서 손가락 끝점 검출 및 건반 누름을 검출해야 한다. 그러나 기존의 방법을 적용하기 어려우므로 새로운 검출 알고리즘 개발이 필요하다.

본 논문의 구성은 다음과 같다. 2장에서는 PC MIDI와 키넥트 센서를 이용한 종이 피아노 기본 시스템의 구성을 살펴보고 3장에서는 키넥트 제어, 건반 인식, 손 영역 검출, 손가락 끝점 검출, 건반 누름 검출 및 MIDI 구동 방법을 설명한다. 4장에서는 실험과 평가를 통해 제안된 방법의 성능을 확인하고 5장에서 결론을 기술하였다.

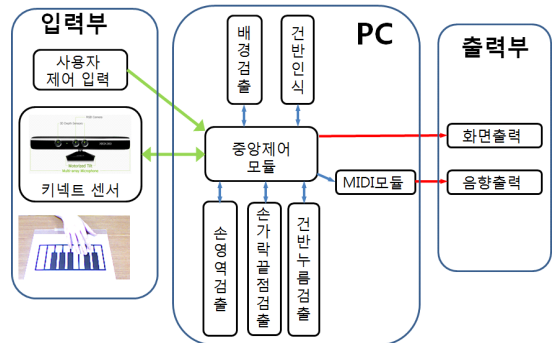


그림 1. 키넥트와 PC MIDI를 이용한 종이 피아노 구성도
Fig. 1. The structure of a paper piano using a Kinect and PC MIDI

II. 종이 피아노 시스템 설계

본 논문에서 제안한 종이 피아노는 그림 1과 같이 크게 입력부, 중앙처리부, 출력부의 3개 모듈로 구성되어 있다.

사용자는 입력부의 키보드 혹은 마우스 입력을 이용하여 종이 피아노 동작을 제어한다. USB 인터페이스로 PC와 연결된 키넥트 센서는 중앙 제어모듈의 동작제어 명령에 따라 초기화하고 종이에 그려진 피아노 건반과 손의 움직임 영상 데이터를 생성하여 PC에 전달한다. 중앙처리부는 전체 동작의 흐름을 제어하는 중앙 제어모듈, 참조 데이터 생성모듈, 건반 누름 인식모듈, 그리고 MIDI 제어모듈로 구성된다.

중앙 제어모듈은 사용자로부터 제어명령어를 입력받아 PC 피아노 프로그램의 키넥트 센서를 초기화하고 설정된 프레임의 크기, 초당 프레임 수에 따라 키넥트 센서로부터 RGB 영상, 깊이 영상을 가져온다. 그리고 사용자 명령에 따라 참조 데이터 생성모듈에서 입력영상으로부터 배경 정보를 작성하고 종이 피아노에 그려진 건반들을 인식하여 그 정보를 메모리에 저장하도록 한다. 또한 사용자의 피아노 연주시작 명령에 따라 건반 누름 인식모듈에서 입력영상을 참조 데이터와 비교하여 손 영역을 검출한 뒤 손가락 끝점들을 추정한다. 손가락 끝점들과 건반인식 정보를 분석하여 눌린 건반을 식별하도록 한다. 건반의 누름 및 떨어짐의 상태가 변할 때마다 MIDI Message를 생성하여 PC MIDI 모듈로 전달한다.

화면출력은 컬러영상, 인식된 건반, 검출된 손 영역, 손가락 끝점을 화면에 출력하며, 음향출력은 MIDI의 출력음을 스피커로 출력한다.

1. 입력부

1.1 키넥트 센서 제어

본 연구에서는 키넥트 센서를 제어하기 위해서 PrimeSense 사의 OpenNI 라이브러리를 이용하였다 [15,16]. 사용된 실행파일과 라이브러리는 표 1과 같다.

표 1. 개발 환경에 사용된 OpenNI 모듈
Table 1. OpenNI modules used in the development environment

항목	모듈
OpenNI Binaries	OpenNI Unstable Build for Windows x86 (32-bit) v1.5.4.0 Development Edition
OpenNI Compliant Middleware Binaries	PrimeSense NITE Unstable Build for Windows x86 (32-bit) v1.5.2.21 Development Edition
OpenNI Compliant Hardware Binaries	PrimeSense Module Unstable Build for Windows x86 (32-bit) v5.1.2.1 Development Edition

본 연구에서는 키넥트 센서에서 생성되는 원시데이터인 컬러영상과 깊이영상 만을 이용하며 OpenNI에서 제공하는 이외 다른 기능은 사용하지 않는다. 이에 Context, DepthGenerator, ImageGenerator 클래스의 객체만 생성, 사용하며 키넥트의 초기화 와 영상데이터 생성을 그림 2와 같이 진행하였다.

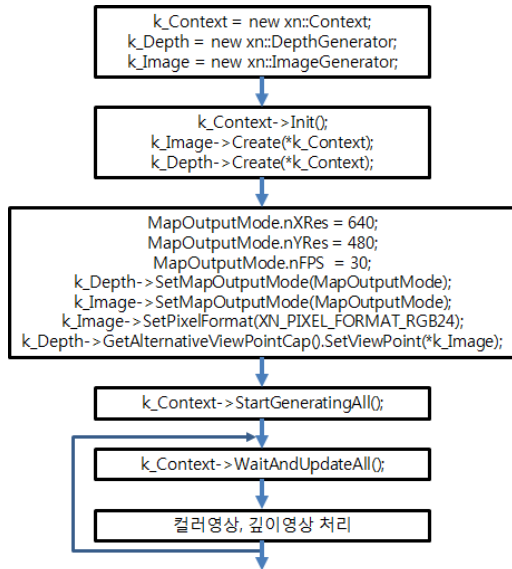


그림 2. 키넥트 센서의 초기화 및 영상데이터 생성 과정
Fig. 2. Kinect sensor initialization and image data generation process

먼저 Context의 멤버함수 init()을 이용하여 초기화하고, DepthGenerator, ImageGenerator의 Create 멤버함수의 매개변수로 전달한다. 그리고 XnMapOutputMode 구조체를 이용하여 DepthGenerator, ImageGenerator 객체의 출력모드를 프레임 크기는 640x480 픽셀, 초당 30프레임으로 설정하였다. 이어서 ImageGenerator 객체의 SetPixelFormat 멤버함수를 이용하여 픽셀값이 RGB 24bit (= 3 x 8bit) 형식으로 저장되도록 하였다. 센서간 거리 차로 인한 깊이영상과 컬러영상의 좌표 불일치를 보정하기 위해서 다음을 수행한다. DepthGenerator 객체의 멤버함수인 GetAlternativeViewPointCap()을 호출하여 AlternativeViewPointCapability 클래스 객체를 반환받는다. 이 객체의 SetViewPoint() 멤버함수에 ImageGenerator 객체의 포인터값을 매개변수로 전달한다.

이상의 과정으로 키넥트 센서의 초기화가 완료되면 Context 객체의 멤버함수인 StartGeneratingAll()을 호출하여 영상데이터 생성을 시작하도록 한다. 이후 중앙제어모듈은 영상데이터가 필요할 때마다 Context 객체의 멤버함수인 WaitAnd-UpdateAll()을 호출하여 새로운 컬러영상과 깊이영상을 요청하고 한다. 그리고 DepthGenerator의 GetDepthMap() 멤버함수를 이용하여 깊이영상 데이터의 시작주소를, ImageGenerator의 GetImageMap() 멤버함수를 이용하여 컬러영상 데이터의 주소를 받아와서 영상처리 과정에 사용한다. 키넥트로부터 수신된 컬러영상과 깊이영상

의 예를 그림 3에 나타내었다. 물체경계에서 깊이영상의 변화 특성을 살펴보기 위해서 (가) 원거리, (나) 손가락 경계가 있는 근거리, (다) 손가락 경계가 없는 근거리 수평선을 임의로 정하고 깊이영상의 변화를 기록하여 그림 4에 나타내었다.

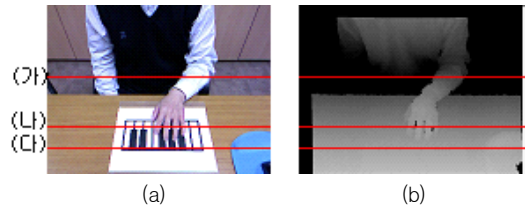


그림 3. 키넥트 센서의 출력영상 (a)컬러영상 (b)깊이영상
Fig. 3. Output Images of a Kinect sensor (a)RGB image (b)depth image

그림 3(b)는 식 (1)을 이용하여 (i,j) 좌표에서의 깊이영상 값 d(i,j)를 흑백영상 값으로 변환한 영상이다.

$$(i,j) = \begin{cases} 0 & , \text{if } d(i,j) = 0 \\ & \text{or } d(i,j) > 800 \\ 255 & , \text{elseif } d(i,j) < 400 \\ 255 * \frac{800 - d(i,j)}{400} & , \text{else} \end{cases} \quad (1)$$

$$(i,j) = G(i,j) = B(i,j) = d(i,j)$$

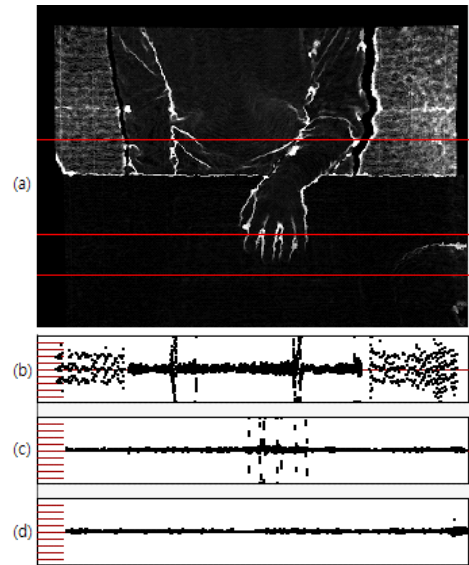


그림 4. 깊이영상 시변성 (a)최대변화 영상 (b) 3-(가)에서의 변화 (c) 3-(나)에서의 변화 (d) 3-(다)에서의 변화
Fig. 4. Time variant depth image (a) image of maximum variation, (b),(c),(d) depth variation on line 가, 나, 다 of Fig. 3

그림 4(a)는 물체 경계에서 깊이영상 값의 시변특성을 살펴보기 위하여 그림 3의 부동산자제로 연속 촬영된 200 프레임에 대해서 식 (2)를 이용하여 변환한 영상이다.

$$i, j) = \max d(i, j) - \min_n d_n(i, j), 1 \leq n \leq 200 \quad (2)$$

$$\Delta i, j) = \min(50, D(i, j)) \times 50$$

$$R(i, j) = G(i, j) = B(i, j) = \Delta(i, j)$$

그림 4(b), (c), (d)는 그림 3의 (가), (나), (다) 높이에 서 가로축 상에 존재하는 각 픽셀에서의 깊이영상 측정값과 평균값과의 차인 en(i,j)를 표시한 그림이다. 수직축의 단위 눈금은 5mm 거리를, 수평축은 x좌표인 i를 나타내고 있다.

$$(i, j) = d_n(i, j) - m(i, j), 1 \leq n \leq 200 \quad (3)$$

$$m(i, j) = \frac{1}{200} \sum_{n=1}^{200} d_n(i, j)$$

그림 4에서 알 수 있듯이 키넥트 센서의 깊이영상 값은 단일 평면일 경우 시변값은 ±3 이내이지만, 거리가 멀어질수록 깊이영상 값의 시변성이 커지며, 특히 물체의 경계에서 변화 폭이 크게 증가함을 보이고 있다. 그러므로 깊이영상만으로 손의 경계점을 추정하는 것은 오류를 포함하기 때문에 이에 대한 보완이 필요함을 알 수 있다.

1.2 사용자 제어 입력

MFC 클래스로 만들어진 대화상자와 컨트롤 기능을 사용하여 구현하였다. 사용자는 버튼을 이용하여 먼저 키넥트 장비를 초기화한다. 만일 키넥트의 초기화 과정에 오류가 발생하면 다른 버튼들을 비활성화 시킨다. 그리고 초기에 설정된 깊이영상의 범위값을 이용하여 컬러영상 화면에 건반인식 후보영역을 표시해주고, 필요시 사용자가 마우스를 이용하여 재설정하도록 하였다. 다음 단계로서 사용자는 버튼을 이용하여 배경화면 정보를 검출하고 깊이영상의 설정 범위값으로 건반 인식을 수행하여 종이 피아노 수행을 위한 기초정보를 작성한다. 초기화가 완료되면 종이 피아노 연주시작 버튼을 활성화하고 사용자의 시작 명령을 받게되면 쓰레드를 생성하여 연주 상태로 진입한다. 사용자는 연주중지 버튼을 이용하여 쓰레드를 중지할 수 있도록 하였다.

2. 종이 피아노 PC 구현부

2.1 배경 검출

이 모듈은 입력영상으로부터 관측영역을 설정하기 위한 배

경 마스크를 작성하고, 피아노 연주모드에서 입력영상으로부터 손 영역을 검출하기 위한 배경의 평균값이 정보를 검출한다. 먼저 깊이영상을 입력받아 설정된 깊이 범위에 속하는 픽셀들을 1, 범위 밖에 속하는 픽셀들을 0으로 하는 2진 마스크를 작성한다. 이전 프레임에서 작성된 2진 마스크와 현재 프레임의 마스크 값이 다른 픽셀 수가 일정값 이상이면 움직임이 있는 영상으로 간주하여 배경 검출 초기단계로 복귀한다. 연속 10개 프레임에서 변화가 일정값 미만이면 고정된 영상으로 간주하여 배경 검출을 시작한다. 배경 검출은 연속 촬영된 100 프레임의 깊이영상에 대해서 식 (4)를 이용하여 배경 깊이영상 bD(i,j)와 2진 배경 마스크 bM(i,j)를 구한다.

$$m(i, j) = \begin{cases} u(d_n(i, j)), & \text{단 } u(d) \text{는 단위계단함수} \\ 0, & \text{if } m(i, j) = 0 \end{cases} \quad (4)$$

$$bD(i, j) = \begin{cases} \sum_{n=1}^{100} d_n(i, j), & \text{otherwise} \\ m(i, j) \end{cases}$$

$$bM(i, j) = \begin{cases} 0, & n_dist1 \leq bD(i, j) \leq n_dist2 \\ 1, & \text{otherwise} \end{cases}$$

2.2 건반 패턴 인식

배경 검출이 완료된 후 실행되는 건반 패턴 인식 단계에서는 피아노 연주모드에서 건반 식별을 위해 필요한 건반 패턴 정보와 사용자의 종이건반 누름 여부를 판정하기 위한 건반값이 정보를 검출한다. 종이에 그려진 피아노 흰 건반은 이진화 과정을 통하여 해당 영역을 하양계, 검은 건반과 주변 테두리를 검게 변환함으로써 쉽게 검출 할 수 있다.

건반 패턴 정보를 생성하기 위해 먼저 배경검출에서 구한 배경 마스크 bM(i,j)와 컬러영상 rgbImage(i,j)를 AND 연산하고, 문턱치를 이용하여 컬러영상을 흑백 2진 영상으로 변환한다. 그리고 외곽선을 검출한 뒤, 흰 건반 면적의 허용 범위값을 이용해 흰 건반 영역들을 분리한다. 검출된 흰 건반 외곽선 각각의 가로축 평균 좌표값을 구한 뒤, 이 값들의 정렬 순서에 따라 건반 번호를 부여한다. 이후 각각의 흰 건반 영역내 좌표에 건반 번호를 저장함으로써 건반 패턴 정보 KeyPattern(i,j)를 생성할 수 있다.

그리고 건반 패턴 정보는 KeyD(i,j)는 식 (5)를 이용하여 생성한다.

$$eyD(i, j) = u(KeyPattern(i, j)) \times bD(i, j) \quad (5)$$

, 단 u(d)는 단위계단함수

검은 건반은 흰 건반들 사이에 위치하며 그 쪽은 흰 건반의 폭과 비례관계에 있다. 검은 건반 검출을 위해서 먼저 이진화된 영상을 역상으로 변환한 뒤 앞에서 검출된 흰 건반들을 포함하는 최외각 사각형 영역으로 검색 영역을 제한한다. 그리고 일정 너비 이하의 흰색 경계선들을 제거한 뒤, 흰건반 추출과정을 적용하여 검은 건반 패턴 정보와 건반깊이 정보를 검출한다. 이렇게 구한 흰 건반, 검은 건반의 정보를 합하여 전체 건반 패턴 정보를 완성하였고 그림 5에 인식된 건반을 나타내었다.



그림 5. 종이건반 패턴 인식 결과
Fig. 5. Recognition result of paper keyboard pattern

2.3 손 영역 검출

건반 패턴 인식이 완료되면 사용자의 제어입력에 의해 피아노 연주모드로 진입한다. 먼저 배경 검출 과정에서 생성된 평균 2진 배경 마스크 $bM(i,j)$ 와 입력된 깊이영상을 곱하여 관측영역을 설정한다. 그리고 관측영역내 각 픽셀에서 입력 깊이값이 배경 깊이값보다 작을 경우 손 영역에 속한다고 1차 판정한다. 이 경우 1.1절에서 살펴본 바와 같이 깊이영상의 시변특성, 물체경계면에서의 오류로 인해 실제 손 영역과 차이가 발생한다. 그리고 손 영역이 종이건반과 접촉하게 될 경우, 손가락 영역을 분리할 때 접촉된 부분이 다소 잘려나가는 문제가 발생하게 된다. 이로 인해 몇 픽셀 정도의 손가락 끝의 오차가 생기게 된다. 그러므로 컬러영상을 YCbCr 공간으로 변환한 뒤, 식 (6)의 피부 색상 판정 기준을 이용하여 1차에서 구한 손 영역을 중심으로 영역검출을 다시 실행한다[9].

$$inMap = \frac{C - C_r}{2} + \left| \frac{C_b - C_b}{2} \right| \quad (6)$$

그림 6(a)에 컬러영상, 6(b)에 깊이영상만 이용해서 검출된 손 영역, 6(c)에 컬러영상으로 보완된 손 영역을 나타내었는데 컬러를 이용함으로써 검출오류가 감소하였음을 알 수 있다.

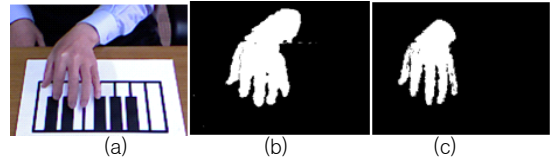


그림 6. 손 영역 검출 (a) 컬러영상 (b) 깊이영상만 이용 (c) 깊이영상과 피부색상 이용
Fig. 6. Decision of hand (a) color image (b) using depth image only (c) using depth image with skin color

2.4 손가락 끝점 검출

종이건반을 누르는 동작에서 손가락들은 손 영역의 외곽에 위치한다. Convex hull 검출 알고리즘을 사용하면 외곽에 위치하는 손가락 끝점을 용이하게 검출할 수 있다[17]. Convex hull이란 그림 7과 같이 특정 점들의 집합에서 가장 외곽에 있는 점들을 연결하되 오목함이 없는 외곽선을 의미한다.

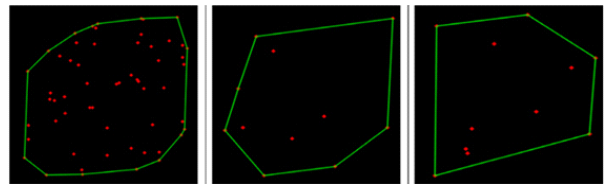


그림 7. Convex hull 검출 예
Fig. 7. Examples of convex hull

손 영역을 포함하는 이진영상 $bHand(i,j)$ 에서 convex hull을 검출하는 알고리즘은 다음과 같다.

1단계: 이진영상의 왼쪽 상단부터 오른쪽 하단까지 순차적으로 이동하면서 손 영역의 왼쪽과 오른쪽의 최외곽점들을 검출하여 $QL(m)$, $QR(n)$ 에 각각 저장한다. 만일 동일 행에서 왼쪽과 오른쪽의 최외곽 좌표가 같으면 $QL(m)$ 에만 해당 좌표를 추가한다.

2단계: QL 로부터 순차적으로 가져온 좌표 $QL(h)$ 와 Q 에 최근에 저장된 2개의 좌표 $Q(k-2)$, $Q(k-1)$ 를 식 (7)에 적용하여 $CCW(Q(k-2), Q(k-1), QL(h)) < 0$ 을 만족하면 $QL(h)$ 를 $Q(k)$ 에 저장한다. 만일 만족하지 않으면 $Q(k-1)$ 를 Q 에서 제거한다.

$$CCW(A, B, C) = (B_y - A_y) \times (C_x - A_x) - (B_x - A_x) \times (C_y - A_y) \quad (7)$$

3단계: QR 로부터 역순으로 가져온 좌표 $QR(h)$ 와 Q 에

최근에 저장된 2개의 좌표 $Q(k-2)$, $Q(k-1)$ 를 식 (7)에 적용하여 $CCW(Q(k-2), Q(k-1), Q_L(h)) < 0$ 을 만족하면 $QR(h)$ 를 $Q(k)$ 에 저장한다. 만일 만족하지 않으면 $Q(k-1)$ 를 Q 에서 제거한다.

4단계: 2, 3단계에서 Q 에 저장된 좌표들이 손 영역의 외곽 정점이 되며 이 정점들이 그림 8(a)와 같이 손가락 끝점 후보가 된다. 그러나 하나의 손가락에 다수개의 정점이 검출될 수 있으므로 일정 거리 이내의 가까운 점이 연속될 경우, 그 점의 가장 하위 포인트만 남기고 나머지는 제거하였다. 그리고 손 영역의 평균높이 상단에 존재하는 외곽정점도 제거하였다.

이상의 과정을 통하여 검출된 손가락 끝점을 손 영역과 같이 그림 8(b)에 나타내었다.



그림 8. (a) convex hull 검출 (b) 손가락 끝점 검출
Fig. 8. (a) convex hull detection (b) finger tip detection

2.5 종이건반 누름 검출

검출된 손가락 끝점 좌표를 이용하여 건반 패턴 정보 $KeyPattern(i, j)$ 에 저장된 값 확인함으로써 손가락 끝이 위치하는 건반을 식별할 수 있다. 그러나 종이건반의 누름 여부를 판별하기 위해서는 추가 정보가 필요하다. 본 연구에서는 손가락 끝점의 깊이값과 건반깊이 정보 $KeyD(i, j)$ 을 비교하여 절대 오차값이 2mm 이내인 경우 해당건반을 눌렀다고 판정하였다. 그러나 끝점 하나만 사용할 경우 누름판정의 오류가 발생하므로 손가락 끝점 주변의 손가락 경계영역에서 누름 여부를 판정하고 다수의 결과를 이용하여 최종 누름여부를 판정하였다.

2.6 MIDI 제어

MIDI 제어모듈에서는 앞에서 검출한 사용자의 종이 피아노 건반 누름 상태를 입력받아 이전 누름상태와 비교한다. 새로이 눌러진 경우는 눌러진 건반번호에 해당되는 음표번호와 채널, 속도, 음표 on 정보로 $MidiOutShortMsg$ 를 생성하고 PC의 MIDI를 제어하여 피아노 음을 출력한다. 이전에 눌러

진 건반에서 손가락이 떨어지면 음표번호와 채널, 속도, 음표 off 정보로 $MidiOutShortMsg$ 를 생성하고 PC의 MIDI를 제어하여 피아노 음을 출력을 중지한다.

3. 화면출력부

제안된 페이퍼 피아노의 구현 결과는 그림 9과 같다. 키넥트로부터 받은 컬러영상을 그림 9의 하단 픽처 컨트롤에 그려 주고, 깊이영상과 컬러영상을 이용하여 인식된 종이 피아노 건반 패턴, 검출된 손 영역, 그리고 손가락 끝점을 상단 픽처 컨트롤에 표시하였다. 그리고 건반 누름 인식결과를 왼쪽 가상건반에 표시하였다.

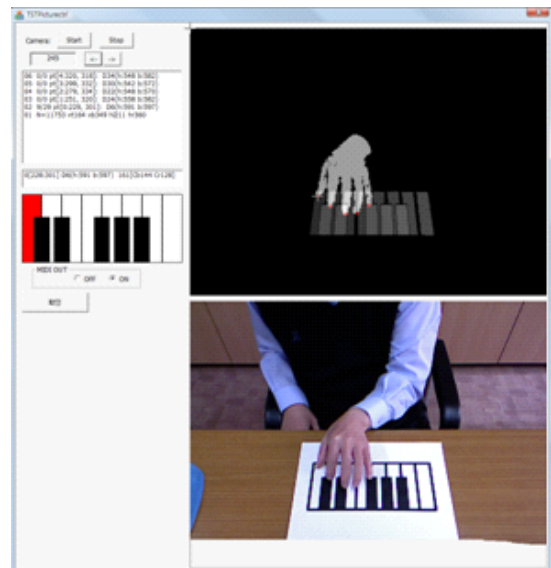


그림 9. 키넥트를 이용한 페이퍼 피아노 구현결과
Fig. 9. Result of Paper piano implementation with Kinect

가상건반은 1옥타브의 건반 영역을 나타내고 있으며, 종이 건반의 현재 누름 상태를 표시한다. 매 프레임마다 건반 누름을 인식하고 결과에 따라 누른 건반을 붉은 색으로, 누르지 않은 건반을 흰색으로 표시하게 된다.

III. 실험 및 결과

본 연구에서 제안한 내용을 바탕으로 직접 software를 구현하고 키넥트와 연동 실험을 통하여 성능 분석을 수행하였다. Software는 Windows 7환경에서 NET framework 2.0과 MFC Dialog기반으로 제작하였고 XBOX360 키넥트

를 사용하였다. Kinect는 컴퓨터 모니터 상단에 설치하고 50cm 떨어진 거리에서 45도 아래에 위치한 종이건반을 촬영하도록 하였다. 키넥트의 출력모드를 프레임 크기를 640x480 픽셀, 초당 30프레임으로 설정하였다.

제안된 알고리즘의 성능평가를 위해서 한순간 하나의 건반만 누르는 동작과 두 개의 건반, 세 개의 건반을 동시에 누르는 영상을 순차적으로 연속 촬영하고 컬러영상과 깊이영상을 파일로 저장하였다. 그리고 수작업으로 영상의 매 프레임에서 건반 누름 여부와 누른 건반 정보를 작성하였다. 이후 제안된 알고리즘을 이용하여 검출된 건반 누름 정보와 비교하여 정확도를 평가하였다.

종이건반 피아노의 건반 1개만 누르는 연속 영상은 모두 600 프레임이며 수작업 확인결과 이 중 178 프레임에서 건반 누름이 있었다. 표 2에 알고리즘에 의한 건반 누름 검출결과를 나타내었다. 누름이 없는 422 프레임 중에서 7 프레임의 경우 오검출이 발생하여 정확도는 98%였다. 누름이 있는 178 프레임 중에서 15 프레임의 경우 검출오류가 발생하여 정확도는 90% 성공함으로써 전체 정확도는 96%를 보였다

표 2. 종이건반 피아노 1개 건반 누름 검출 결과
Table 2. Detection result of touching 1 key on paper keyboard

검출건반수 입력건반수	0	1	2	3	4	5	각 케이스별 이미지 횟수	정확도 (%)
0	415	7	0	0	0	0	422	98.3
1	15	161	2	0	0	0	178	90.4
전체							600	96.0

표 3. 종이건반 피아노 2개 건반 누름 검출 결과
Table 3. Detection result of touching 2 keys on paper keyboard

검출건반수 입력건반수	0	1	2	3	4	5	각 케이스별 이미지 횟수	정확도 (%)
0	342	8	0	0	0	0	350	97.7
1	3	38	0	0	0	0	41	92.7
2	0	32	277	0	0	0	309	89.6
전체							700	93.9

표 4. 종이건반 피아노 3개 건반 누름 검출 결과
Table 4. Detection result of touching 3 keys on paper keyboard

검출건반수 입력건반수	0	1	2	3	4	5	각 케이스별 이미지 횟수	정확도 (%)
0	203	6	0	0	0	0	209	97.1
1	0	12	0	0	0	0	12	100
2	0	3	9	0	0	0	12	75.0
3	0	13	53	270	0	0	336	80.4
전체							569	86.8

종이건반 피아노의 건반 2개를 동시에 누르는 연속 영상은 모두 700 프레임이며 수작업 확인결과 이 중 41 프레임에서 건반 1개 누름, 309 프레임에서 건반 2개 누름이 있었다. 표 3에 알고리즘에 의한 건반 누름 검출결과를 나타내었다. 누름이 없는 400 프레임의 검출 정확도는 98%이고 1개의 누름이 있는 41 프레임의 검출 정확도는 92.7%, 2개의 누름이 있는 309 프레임에서는 89.6% 성공함으로써 전체 정확도는 93.9%를 보였다.

종이건반 피아노의 건반 3개를 동시에 누르는 연속 영상은 모두 569 프레임이며 수작업 확인결과 이 중 12 프레임에서 건반 1개 누름, 12 프레임에서 건반 2개 누름, 336 프레임에서 건반 3개 누름이 있었다. 표 4에 알고리즘에 의한 건반 누름 검출결과를 나타내었다. 누름이 없는 209 프레임의 검출 정확도는 97.1%이고 1개의 누름이 있는 12 프레임의 검출 정확도는 100%, 2개의 누름이 있는 12 프레임에서는 75%, 3개의 누름이 있는 336 프레임에서는 80.45% 성공함으로써 전체 정확도는 86.8%를 보였다

V. 결론

본 논문에서는 키넥트의 3차원 영상정보를 이용하여 손가락 움직임을 인식하고 이 정보를 이용하여 종이건반 피아노를 구현하는 방법을 제안하였다. 키넥트는 일반적인 카메라 장치와 달리 조명과 같은 외부 요인에 영향을 적게 받으며, 저가로 장치이지만 비교적 신뢰할 수 있는 3차원 영상정보를 제공한다. 키넥트의 컬러영상과 깊이영상을 이용함으로써 건반 패턴 인식, 손 영역, 손가락 끝점, 종이건반 누름 검출의 정확도를 향상시킬 수 있었다. 이 정보를 이용하여 피아노 연주 입력장치를 구현함으로써 PC기반 피아노 연주시스템 구현의 편리성을 보였다.

이상과 같이 구현된 종이건반 피아노 기능을 활용하여 악

기 초보자가 PC 기반 피아노 연주 학습에 활용할 수 있는 가능성을 확인하였다. 이를 통해 저학년의 초등학생들이 음악과 악기에 대한 이해 및 접근성을 높이고, 창의성과 음악적 능력을 향상에 기여하고자 한다.

본 논문에서 제시한 방법은 그 가능성을 제시하기 위한 것으로 아직 보완이 필요하다. 향후 손가락 끝점 검출의 정확도 향상을 위한 연구를 통해 보완작업을 진행할 예정이다.

VI. 감사의 글

이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2012-0004681)

참고문헌

- [1] Akai EWI4000S, <http://www.akaipro.com/>
- [2] WX5 YAMAHA Midi Controller, <http://www.yamaha.com/>
- [3] Finale&SmartMusic, <http://www.makemusic.com/>
- [4] P. Messic, "Maximum MIDI: Music Applications in C++," Manning Publications, 1998.
- [5] 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.
- [6] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *IEEE Conference on Computer Vision and Pattern Recognition*, vol.1, pp.511-518, Dec. 2001.
- [7] B.M. Kim, J.W. Kim, and K.H. Lee, "An Application of AdaBoost learning Algorithm and Kalman Filter to Hand Detection and Tracking," *Journal of The Korea Society of Computer and Information*, vol.10, no.4, pp.47-56, Sep. 2005.
- [8] 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.
- [9] 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.
- [10] Microsoft Kinect, <http://www.microsoft.com/en-us/kinectforwindows/>
- [11] 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.
- [12] S.H. Lee, H.S. Hahn, and Y.J. Han, "The Estimation of Hand Pose Based on Mean-Shift Tracking Using the Fusion of Color and Depth Information for Marker-less Augmented Reality," *Journal of The Korea Society of Computer and Information*, vol.17, no.7, pp.155-166, July 2012.
- [13] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.17, no.8, pp.790-799, Aug. 1995.
- [14] kinect piano download, http://www.google.co.kr/#q=kinect+piano+download&hl=ko&newwindow=1&sa=N&tbo=u&source=univ&tbm=vid&ei=RCPQUNz8OOqaiAe1qoHAAG&ved=0CEsQqwQ4Cg&bav=on.2,or.r_gc.r_pw.r_qf.&bvm=bv.1355534169,d.aGc&fp=7605525a9eab9576&bpcl=40096503&biw=1090&bih=1036
- [15] OpenNI, <http://www.openni.org/documentation/>
- [16] OpenNI, <http://www.openni.org/Downloads/OpenNI-Modules.aspx>
- [17] Xianquan Zhang and Zhenjun Tang, "A Fast Convex Hull Algorithm for Binary Image," *Informatica* vol.34, no.3, pp.369-376, Mar. 2010.

저 자 소 개



이 정 철

1984년 : 서울대학교
전자공학과 공학사
1988년 : 서울대학교
전자공학과 공학석사
1998년 : 서울대학교
전자공학과 공학박사
1985년~2000년: ETRI 책임연구원
2000년 : L&H Korea 전문위원
2001년 : (주)보이스텍 전문위원
2002년 : (주)코난테크놀로지
책임연구원
2002년 9월~현재: 울산대학교
전기공학부 부교수
관심분야 : 디지털신호처리,
음성신호처리, 음성합성
Email : jungclee@ulsan.ac.kr



김 민 성

2006~현재: 울산대학교
컴퓨터정보통신공학부
학사과정
관심분야: 디지털신호처리
Email : ms0304@ulsan.ac.kr