

체감형 콘텐츠 개발을 위한 연속동작 매칭 방법에 관한 연구

이형구

한국산업기술대학교 게임공학과

hgl@kpu.ac.kr

A Study on the Gesture Matching Method for the Development of Gesture Contents

HyoungGu LEE

Dept. of Game, Korea Polytechnic University

요 약

본 연구에서는 윈도우즈 PC용 연속동작 감지 카메라, Xtion을 이용한 PC-윈도우 플랫폼 기반의 연속동작 녹화 및 매칭방법의 개발 내용을 소개한다. 해당 방법을 개발하기 위해 카메라를 통해 얻은 깊이 정보, RGB 화상 정보, 뼈대 정보를 가공하고 비교하는 API를 먼저 개발하였다. 유효관절만을 선택적으로 비교하는 pose 비교 방법이 개발되었으며, 연속동작 비교에서는 pose 사이에 다른 틀린 pose가 섞여도 인식할 수 있는 방법이 개발되었다. 특정 pose나 연속동작 검출을 위해 샘플 데이터를 기록하고 테스트할 수 있는 도구가 개발되었다. 6종류의 다른 pose 및 연속동작을 촬영하고 테스트한 결과, pose는 100%의 인식성공과 연속동작은 99%의 인식성공이 이루어져 개발된 방법의 유용성을 검증할 수 있었다.

ABSTRACT

The recording and matching method of pose and gesture based on PC-window platform is introduced in this paper. The method uses the gesture detection camera, Xtion which is for the Windows PC. To develop the method, the API is first developed which processes and compares the depth data, RGB image data, and skeleton data obtained using the camera. The pose matching method which selectively compares only valid joints is developed. For the gesture matching, the recognition method which can differentiate the wrong pose between poses is developed. The tool which records and tests the sample data to extract the specified pose and gesture is developed. 6 different pose and gesture were captured and tested. Pose was recognized 100% and gesture was recognized 99%, so the proposed method was validated.

Keywords : gesture detection (연속동작 감지), pose (자세), gesture matching (연속동작 매칭), skeleton (뼈대)

Received: Dec. 06, 2013 Accepted: Dec. 16, 2013

Corresponding Author: HyoungGu LEE(Korea Polytechnic Univ.)

E-mail: hgl@kpu.ac.kr

© The Korea Game Society. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

ISSN: 1598-4540 / eISSN: 2287-8211

1. 서론

현재 연속동작 감지 카메라를 이용한 체감형 콘텐츠는 대부분 게임에 한정되어 있으며, 그것 또한 Microsoft사의 XBOX360 콘텐츠에 한정되어 있다. 본 연구에서는 특정회사의 라이선스 제약을 받지 않는 카메라를 이용한 연속동작 매칭 기술 및 API(Application Programming Interface)를 개발함으로써 그 범위를 일반 PC로 확장시키고, 게임을 제외한 다른 유용한 콘텐츠에서도 깊이 카메라를 사용할 수 있는 기반을 마련하려고 한다.

서술된 내용은 [Fig. 1]에 나와 있는 Microsoft사의 Kinect[1,2]와 동일한 성능을 가지는 PC용 카메라 단말인 ASUS사의 Xtion[3,4] 제품을 활용한 카메라 제어분석 API를 먼저 개발하고 이를 바탕으로 콘텐츠 개발에 적용할 수 있는 연속동작 녹화 및 매칭 방법을 개발한 내용이다.



KINECT
for XBOX 360.

(a) Kinet of Microsoft



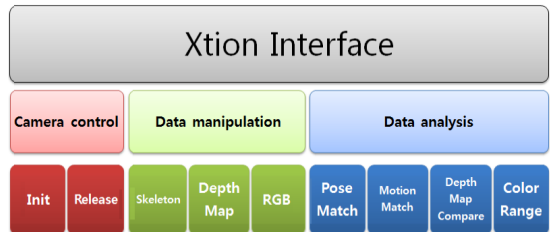
(b) Xtion of ASUS

[Fig. 1] Gesture cameras of Microsoft and ASUS

콘텐츠 개발 시 PC의 경우 일반 게임 플랫폼과 다르게 하드웨어 장비의 호환성에서 문제가 생긴다. 고정된 사양의 플랫폼이 아니라 운영체제의 업데이트와 디바이스 드라이버의 업데이트 등 계속해

서 사양이 변하기 때문에, 그에 따라 개발환경이 계속해서 변화하기 마련이다.

본 연구에서는 연속동작 감지 카메라 SDK와 일반 어플리케이션 사이에서, 변화하는 개발 환경으로부터 자유로운 카메라를 제어하는 API를 개발, 공급하여 PC의 연속동작 감지 콘텐츠 개발 장벽을 낮추어 좀 더 수월하게 콘텐츠를 개발할 수 있는 환경을 구성하고자 하였다. [Fig. 2]는 Xtion 인터페이스를 기반으로 연속동작 녹화 및 검출 응용프로그램에 사용되는 카메라 API 구성도를 보여준다.

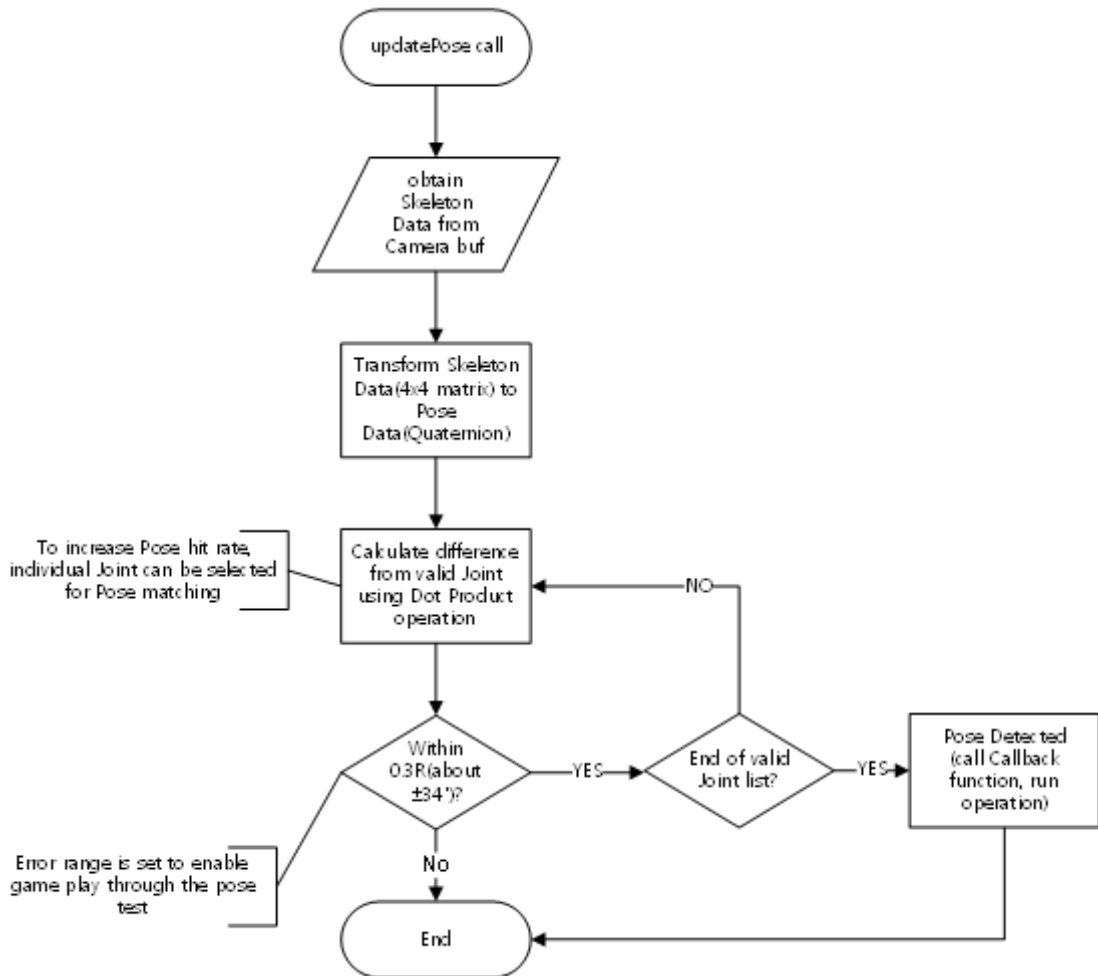


[Fig. 2] Camera API configuration

본 논문의 구성은 다음과 같다. 2장은 개발된 카메라 제어 인터페이스 부분에 대한 설명이고 3장에서는 이를 바탕으로 개발된 동작 녹화 및 매칭 테스트 도구 및 이에 대한 실험 내용을 설명하고 4장은 결론 및 향후 연구방향이다.

2. 카메라 제어 인터페이스

카메라 제어 인터페이스는 카메라의 raw data를 제공하는 OpenNI[5,6] 라이브러리와 RGB 화상이나 깊이 화상을 분석 가공하는데 사용되는 OpenCV[7]를 결합하여 구현하였다. [Table 1]은 구현된 API의 기능을 분류별로 보여준다. [Fig. 3]은 pose 동작의 검출 및 매칭 방법이 구현된 순서도이고 [Fig. 5]는 연속동작 검출 및 매칭 방법이 구현된 순서도이다.



[Fig. 3] Flow chart of the pose detection and matching

2.1 Pose 검출

[Table 1]의 ‘Manipulation’ 분류에서 ‘Compare the acquired skeleton info with the pose data previously saved’ 기능은 [Fig. 3]의 순서도와 같이 정지 자세를 비교한다. 먼저 카메라에서 얻은 현재 정지자세의 skeleton 데이터(4*4 행렬로 OPENNI에서 제공하는 데이터)를 pose 데이터 (Quaternion)로 변환시키고 저장된 각 관절부의 Quaternion과의 각도 차이를 계산한다.

[Table 1] Implemented camera API function

class	function
Initialize	Initialize the camera and connect
	Display the information of multiple connected cameras
Information	Obtain the depth info of display which is being captured by camera
	Obtain the RGB info of display which is being captured by camera
	Obtain the user position at the space
	Obtain the hand point of user
Manipulation	Compare the acquired skeleton info with the pose data previously saved
	Make animation by matching skeleton info and skeleton
Analysis	Accumulate the skeleton info and match with the specified animation
	Based on user info, acquire the user shape through the RGB image
	Obtain the specified color rate in the RGB image
	Calculate the difference between the specified depth info and present depth info

계산된 차이가 유효범위 이내이면 동일한 pose 로 인식한다. 이때 유효 각도는 미리 저장된 pose 데이터와 실시간으로 갱신되는 사용자의 관절 회전 값을 비교할 때 각 관절의 매치 여부를 판단하는 기준값으로서 상수값으로 그 범위를 지정한다.

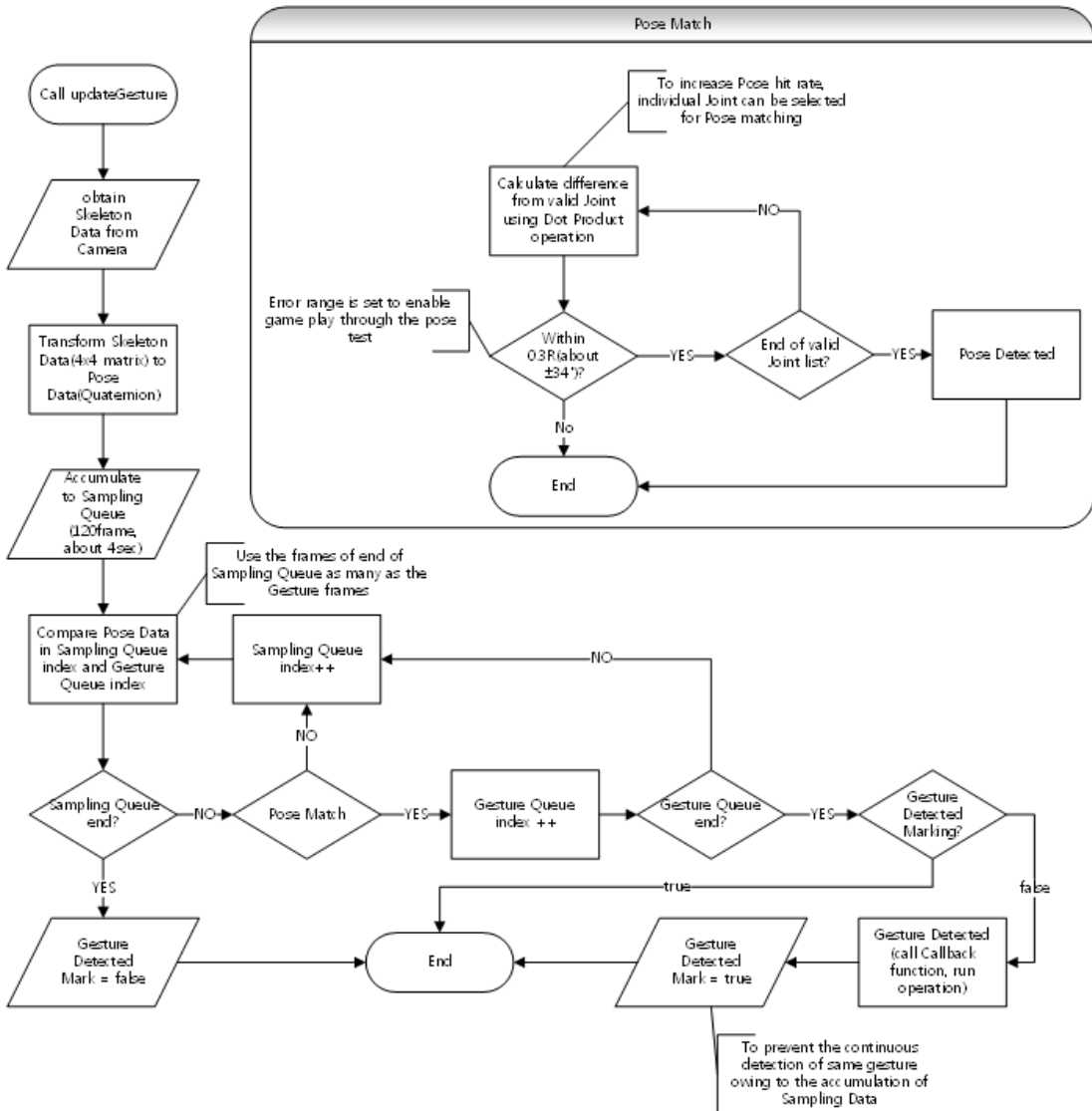
미리 저장된 pose 데이터는 실시간으로 획득하는 skeleton 데이터와 동일한 정보(각 관절의 회전 값)와 각 관절별로 비교를 진행할 것인지에 대한 유효관절 여부 정보가 포함된다.

이를 단계별로 다시 설명하면 다음과 같다.

1. 카메라에서 skeleton 데이터를 얻는다.
2. skeleton 데이터를 pose 데이터 형식으로 변환한다.
3. 변환된 현재 데이터와 등록된 pose 데이터를 조인트 단위로 비교한다. 이때 유효관절 여부가 true인 조인트만 비교한다. 모든 관절에 대해 비교를 진행하면 적중률이 떨어지므로, 해당 pose에서 중요한 관절에 대해서만 비교를 수행하도록 구성하였다. 이때 비교에 사용되는 관절을 유효관절이라고 한다.
OPENNI가 제공하는 skeleton 정보는 24개 관절의 회전값을 제공하지만, 예를 들어 오른팔을 이용한 펀치 pose를 검출할 때는 오른팔 어깨, 팔꿈치, 오른팔 손목의 3개에 대해서만 비교를 진행한다.
4. 모든 유효관절의 오차값이 허용범위 이내면 'Pose Detected' 판정 후 후처리 콜백함수를 호출하고, 1개라도 오차 한계 값을 벗어나면 무시하고 다음 검출을 시도한다. 여기서 후처리 콜백함수는 pose 매칭이 성공적으로 수행됐을 때 프로그램이 해야 하는 동작을 정의한 함수이다. 예를 들어 오른팔 펀치가 매치되었을 경우 '키보드의 스페이스바가 입력된 것으로 한다'등의 후처리 작업을 의미한다. pose 데이터에 정의되어 있는 pose의 명칭이 문자열 인자로 전달되며, 이 인자를 비교하여 원하는 동작에 대한 처리를 구현하게 된다.

	1	2	3	4	5	...	109	110	111	112	113	114	115	116	117	118	119	120
S								A	B	C	D	E	F	G	H	I	J	K
G	A	D	H	K														

[Fig. 4] Sampling queue(S) and saved gesture queue(G)



[Fig. 5] Flow chart of Gesture detection and matching

2.2 연속동작 검출

가공 분류 중 뼈대 정보를 누적하여 특정 애니메이션(연속동작)과 매칭 비교하는 기능은 [Fig. 5]의 순서도 흐름으로 수행된다.

연속동작 검출의 경우는 앞에서 설명한 pose 검출 로직을 기반으로, 대전액션게임의 액션 커맨드 입력(중간에 다른 입력이 들어오더라도 흐름상 연

속된 조작이 순서대로 들어오면 올바른 입력으로 인식하고 액션 수행) 형태를 사용하여 검출한다.

먼저 현재 카메라에서 획득한 skeleton 데이터를 pose 데이터로 변환한 뒤 연속동작 매니저의 샘플링 큐(최대 120개, 4초 분량 저장)에 저장한다.

이후 샘플링 큐에서 pose를 가져와 저장된 연속동작 리스트의 첫 번째 값과 샘플링 큐의 pose가

매칭이 되면 연속동작 리스트의 첫 데이터를 pop 시킨다.

저장된 연속동작 리스트가 비게 되면 매칭 됨으로 처리하고, 샘플링 큐가 빌 때까지 저장된 연속동작 리스트가 남아 있으면 비매칭으로 처리한다. 이를 단계별로 다시 설명하면 다음과 같다.

1. 카메라에서 skeleton 데이터를 얻는다.
2. skeleton 데이터를 pose 데이터의 연속으로 변환하여 샘플링 큐를 구성한다.(120 프레임)
3. 샘플링 큐에서 비교할 동작 데이터의 프레임 수 만큼을 떼서 저장된 연속동작 큐와 비교한다.
4. [Fig. 4]와 같이 구성되는 샘플링 큐(S)와 저장된 연속동작 큐(G)의 pose 데이터를 아래와 같이 비교한다.

저장된 연속동작 큐(G)는 연속동작 데이터에서 맨 앞과 맨 뒤, 그리고 프레임스킵에 의한 유효 프레임으로 구성한다.

[Fig. 4]의 각 칸에 표시된 알파벳 문자는 연속동작을 구성하는 특정 pose의 형태를 의미한다.

붉은색 칸은 현재 카메라에서 획득한 연속동작 데이터의 샘플링 큐(S), 푸른색 칸은 저장된 연속동작 데이터를 기반으로 생성한 연속동작 큐(G)를 나타낸다.

[Fig. 4]의 4장의 유효프레임으로 구성된 저장된 연속동작 큐를 검출하는 경우는 다음과 같다.

- ① 샘플링 큐(S)의 110번 프레임과 연속동작 큐(G)의 1번 프레임을 비교한다. (매칭 성공)
- ② 샘플링 큐(S)의 111번 프레임과 연속동작 큐(G)의 2번 프레임을 비교한다. (매칭 실패)
- ③ 샘플링 큐(S)의 112번 프레임과 연속동작 큐(G)의 2번 프레임을 비교한다. (매칭 실패)
- ④ 계속 반복하여 샘플링 큐(S)의 마지막까지 비교한다.
- ⑤ 저장된 연속동작 큐(G)의 모든 pose가 순서

대로 검출되었을 경우 'Gesture Detected'로 판정.

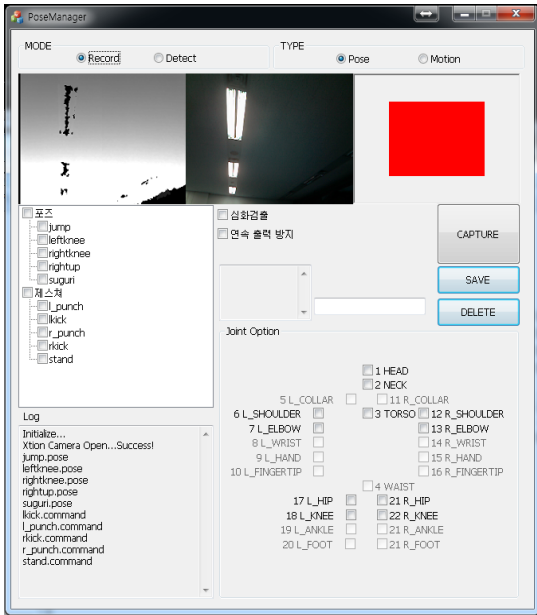
5. 'Gesture Detected'로 판정되면 연속동작이 이전에 검출됐는지 확인하고 현재 처리되고 있는 샘플링 큐에 검출됐음을 마킹한다. 이때 연속동작 검출 시 불러지는 후처리 함수도 호출한다. 'Gesture Detected'의 경우 샘플링 큐의 마지막 구간 내에서 저장된 연속동작 큐의 pose가 모두 검출되면 그 지점까지의 비교를 건너뛰어야 한다. 그 구간 내에서 비교를 반복적으로 수행하면 동일한 연속동작 검출이 연속적으로 일어난다. 이를 방지하기 위해 첫 검출 시 마킹을 사용한다.
6. 해당 연속동작이 검출되지 않을 경우 'Gesture Detected' 마킹을 해제하고, 다음 검출을 시도한다.

3. 연속동작 녹화 및 매칭 테스트

카메라를 이용한 pose 및 연속동작 녹화 및 매칭 테스트 도구는 개발하고자 하는 콘텐츠에서 특정 pose나 연속동작 검출을 위해 샘플 데이터를 기록하고 테스트하는 도구이다.

[Fig. 6]은 pose 및 연속동작 데이터 파일을 생성하고 해당 데이터의 검증을 위해 만들어진 프로그램의 스크린 샷이다. 해당 프로그램이 제공하는 기능은 다음과 같다.

- ① pose 데이터 촬영
- ② 연속동작 데이터 촬영
- ③ 각 데이터 파일의 수정
- ④ 각 데이터 파일의 유효관절 편집
- ⑤ pose 데이터 검출 : 왼쪽 리스트에서 체크박스에 체크 된 데이터를 대상으로 pose 검출을 수행한다.



[Fig. 6] Screen shot of Pose and Gesture matching tool window

- ⑥ 연속동작 데이터 검출 : 왼쪽의 리스트에서 체크박스에 체크 된 데이터를 대상으로 연속 동작 검출을 수행한다.
- ⑦ 심화검출 : 비슷한 동작이 여러 개 등록되어 있을 경우 검출된 모든 동작을 한꺼번에 출력하는 것이 기본이고, 심화 검출 시에는 동일 프레임 내에서 동일한 데이터가 여러 개

검출되었을 경우 가

1

- ⑧ 연속출력 방지 : 원활한 테스트를 위해서 로 그 창에 나타나는 데이터가 연속적으로 같을 경우 출력하지 않는다.

사람의 pose에 대한 아날로그 정보를 디지털 정보로 변환하는 과정에서는 오차가 발생한다. 이 오차는 대상자의 pose 및 연속동작의 비교를 불가하게 만든다.

반복적인 테스트를 통해 실제 인체의 움직임 특성을 분석하고, 특히 카메라에 정면으로 서 있을 경우에는 관절 회전 값의 오차가 심하고 적중률 기여도가 떨어지는 몇 개 관절에 대한 인식을 수행하지 않도록 선택 옵션을 제공하였다.

그 결과 pose 인식율을 콘텐츠 플레이에 방해되지 않는 수준으로 끌어올릴 수 있었고, 왼팔과 오른팔, 팔과 다리 등 각기 다른 관절에 개별적인 포즈를 요구할 수 있게 되었다.

예를 들어 슈팅 게임의 경우 오른팔로 조준을 하면서 왼팔로 발사 제스처를 별도로 내릴 수 있고, 실제 게임에 사용되지 않는 다리관절의 데이터를 처리하지 않게 함으로써 인식 성능의 향상을 가져올 수 있었다.

이외에도 연속동작에서 모든 프레임을 비교하지 않고 특정 몇몇 프레임만 비교하여 판정하는 방법과 관절의 유효각도 설정을 변경하는 것으로 인식

[Table 2] Test result of Pose and Gesture matching

		Pose A	Pose B	Pose C	Average speed	Gesture A	Gesture B	Gesture C	Average speed
A 175cm ~	user1	model	○○○	○○○	below 5ms	model	○○○	○○○	below 30ms
	user2	○○○	○○○	○○○	below 5ms	○○○	○○○	○X○	below 30ms
	user3	○○○	○○○	○○○	below 5ms	○○○	○○○	○○○	below 30ms
	user4	○○○	○○○	○○○	below 5ms	○○○	○○○	○○○	below 30ms
B 170cm ~	user5	○○○	model	○○○	below 5ms	○○X	model	○○○	below 30ms
	user6	○○○	○○○	○○○	below 5ms	○○○	○○○	○○○	below 30ms
	user7	○○○	○○○	○○○	below 5ms	○○○	○○X	○○○	below 30ms
	user8	○○○	○○○	○○○	below 5ms	○○○	○○○	○○○	below 30ms
C 165cm ~	user9	○○○	○○○	model	below 5ms	○○○	○○○	model	below 30ms
	user10	○○○	○○○	○○○	below 5ms	X○○	○○○	○○○	below 30ms
	user11	○○○	○○○	○○○	below 5ms	○○○	○○○	○X○	below 30ms
	user12	○○○	○○○	○○○	below 5ms	○○○	○○○	○○○	below 30ms

를 향상을 가져올 수 있었다.

[Table 2]는 보통 체형을 가진 남자 12명을 대상으로 키를 기준으로 3그룹으로 나누고, 각 그룹의 1명을 촬영자로 지정하여 각기 다른 pose 및 연속동작을 촬영하고 테스트한 결과이다.

각 시험자는 하나의 동작을 3번씩 테스트한다. pose의 경우 카메라 단말 촬영 속도가 30 fps 이므로 최소 33ms의 데이터 지연 시간이 발생하는 것을 고려하였다. 또한 제스처의 경우 지연 시간과 더불어 각 동작의 시작과 끝 pose 검출을 위한 작동 시간이 요구되므로 해당 작동이 끝난 이후의 검출 연산시간을 체크하였다.

4. 결론 및 향후연구

본 연구에서는 카메라 제어분석 API를 바탕으로 연속동작 인식 콘텐츠 개발에 적용할 수 있는 연속동작 녹화 및 매칭 방법을 소개하였다.

본 연구의 연속동작 검출 방법을 적용한 유아용 체감형 콘텐츠가 상용화되어 [Fig. 7]과 같이 전국의 주요 키즈 카페에서 서비스되고 있다.

현재 개발된 콘텐츠 API는 기본적인 카메라 제어 함수와 skeleton, depth, RGB 각각의 데이터를 비교하거나 분석하고, 특정 데이터를 추출하는 함수를 제공한다. 콘텐츠 개발자는 해당 함수를 사용하여 의미 있는 데이터를 얻을 수 있지만, 특수한 목적의 기능을 위해 2차 가공을 해야 하는 경우가 있다. 즉 지정된 2명의 pose를 비교하거나, 2명의 옷 색을 비교하는 경우, 그리고 특정 pose를 지정된 시간에 몇 번 반복하는지 등은 제공된 API로 구현 가능하다. 그러나 이러한 복합 기능을 미리 API 수준에서 제공한다면 개발자는 더욱 짧은 시간에 콘텐츠를 구현하는 것이 가능해진다.

복수개의 카메라를 이용해 캡처한 후 데이터를 병합하는 기능을 추가로 개발하게 되면, 1인 촬영시 발생하는 오차를 많이 줄이고 회전까지도 처리할 수 있을 것으로 예상된다.



[Fig. 7] Applied case of Suggested Gesture detection method

REFERENCES

- [1] Microsoft Kinect, "<http://www.xbox.com/ko-KR/Kinect>"
- [2] C. J. Lim, Park Seung Goo, Jeong Yun Guen, "Development of Motion based Training Contents: "3D Space Exploration" Case Study", Journal of Korea Game Society, Vol. 13, No. 5, pp.63 ~ 72, 2013
- [3] ASUS Xtion, "http://www.asus.com/Multimedia/Xtion_PRO/"
- [4] Sun-Kyung Kang, Sung-Tae Jung, "Development of a Serious Game for Rehabilitation Training using 3D Depth Camera", Journal of Korea Game Society, Vol. 13, No. 1, pp.19 ~ 30, 2013
- [5] OpenNI, "<http://www.openni.org/>"
- [6] Ok Sooyeol, Kim Dalhyun, "Serious Game Design for Rehabilitation Training with Infrared Ray Pen", Journal of Korea Game Society, Vol. 9, No. 6, pp. 151 ~ 161, 2009
- [7] OpenCV, "<http://opencv.org/>"



이 형 구 (LEE, Hyoung Gu)

1993년 2월 서울대학교 계산통계학과 이학사
1996년 2월 서울대학교 대학원 계산통계학과 이학석사
2001년 8월 서울대학교 대학원 컴퓨터공학부 공학박사
2001년 9월-2003년 2월 한국전자통신연구원 영상정보
처리연구팀 선임연구원
2003년 3월-현재 한국산업기술대학교 게임공학과

관심분야 : 인공지능, 영상처리
