



3차원 애니메이션을 위한 손동작 리깅 시스템의 프로토타입 제작 사례

양 정 아 · 박 태 정*

덕성여자대학교 IT미디어공학과

Case Study for Prototype for Hand Motion Rigging in Three Dimensional Animation

Jung Ah Yang · Taejung Park*

Department of Information Technology and Media Engineering, College of Engineering,
Duksung Women's University, Seoul 01369, Korea

[요약]

본 연구에서는 저렴하고 간단하게 제작할 수 있는 스마트 글로브 프로토타입을 제안한다. 제안하는 프로토타입의 테스트를 위해 본 논문에서는 손가락의 편리한 3차원 애니메이션 제작, 손동작 정보 저장 및 전송, 캡처한 손동작의 재현을 위한 로봇팔 제어를 구현하였으며, 구현 방식과 테스트 결과를 논의한다. 이러한 목표를 달성하기 위해서 Arduino로 센서와 로봇, 3ds MAX 간의 통신을 구현하였으며 캡처한 기하 정보는 3ds MAX 환경 하에서 MAXScript를 이용해서 Inverse Kinematics 기반 keyframe 애니메이션으로 변환하였다. 향후 제안하는 방식은 가상 현실 내에서의 정확한 손동작 인식, 인공 지능 기반 수화 인식을 위한 학습 데이터 캡처, 로봇팔 원격 제어 등의 응용으로 확장할 수 있으리라 기대한다.

[Abstract]

We present a simple and cost effective prototype of smart globe in this paper. We have implemented and tested the prototype for efficient three dimensional animation of human fingers, capture, storage, and transmission of finger motions, and controlling robot fingers to replay captured finger motions. With this aim, we have implemented the communication among Arduino, globe sensors, and robot arm. The captured geometry information is transformed to keyframe animation based on the inverse kinematics (IK) technique implemented via MAXScript in the 3ds MAX environment. We expect the proposed method could be used to implement precise recognition of hand positions, capture training data for sign language applications based on artificial intelligence, and control remote robot arms.

색인어 : 3ds MAX, MAXScript, 스마트 글로브, 키프레임 애니메이션, 인버스 키네마틱스

Key word : 3ds MAX, MAXScript, Smart globe, Keyframe animation, Inverse kinematics

<http://dx.doi.org/10.9728/dcs.2019.20.2.271>



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.

Received 20 January 2019; **Revised** 06 February 2019

Accepted 20 February 2019

***Corresponding Author:** Taejung Park

Tel: +82-02-901-8339

E-mail: tjpark@duksung.ac.kr

I. 서 론

1-1 연구 개요 및 배경

3차원 애니메이션 제작 과정과 가상현실(Virtual Reality) 공간 내 애플리케이션에서 사람 손가락의 움직임을 자연스럽고 적절하게 표현하기 위해서는 많은 시간과 노력이 필요하다.

3차원 애니메이션 제작의 경우 인간 신체의 골격에 해당되는 구조(예. 3ds MAX에서의 biped, 그림 1)의 동작을 역운동학(Inverse Kinematics)[1] 또는 모션 캡처(Motion capture)[2] 등으로 구현한 후에 피부나 의상에 해당되는 mesh를 적절하게 입히는 skinning 과정을 거치게 되는데 손가락의 경우 골격 구조 사이의 간격이 좁을 뿐만 아니라 관절 개수도 다른 신체 구조에 비해 상대적으로 많은 특징을 가지고 있다(그림 1). 이러한 조건 하에 3차원 컴퓨터 그래픽스와 애니메이션의 수준이 향상되고 기대치가 높아짐에 따라 인간의 행동, 특히 자연스러운 의사소통 등을 위해 손가락의 자연스러운 미묘한 움직임을 정확하게 표현할 수 있어야 하지만 전술한 구조적인 복잡함 때문에 애니메이터들이 수작업으로 모든 작업을 처리하기 위해서는 많은 시간과 노력이 소요된다. 또한 가장 정확하게 손가락의 움직임을 캡처할 수 있는 스마트 글로브[3]는 여전히 높은 가격 때문에 접근이 어려우며 비교적 저렴한 적외선 기반 센서(예. LeapMotion[4] 또는 Kinect[5])는 조명 조건에 민감[6]하고 오른손과 왼손이 교차되는 특정한 동작에서 정확성이 떨어지는 단점이 있다.

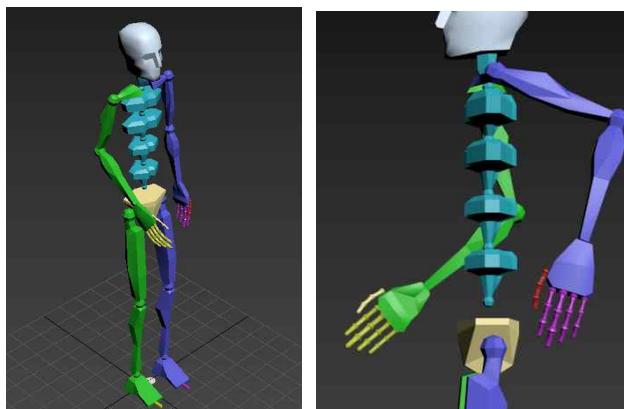


그림 1. 3ds MAX의 캐릭터 골격 구조인 biped(왼쪽), biped의 손가락 구조(오른쪽)

Fig. 1. The biped structure which is the common skeleton for character animation (left), the finger bones of the biped (right)

여러 가상현실 애플리케이션에서도 가상공간 내에 사용자 손가락의 움직임을 캡처해야 하는 상황이 빈번하게 발생한다. 즉, 가상현실 공간에서는 사용자가 자신의 손을 직접 볼 수 없기 때문에 자신의 손과 손가락의 움직임을 감지해서 적절하게 표현해야 하는 필요가 자주 발생한다. 김은솔 외 연구자들이 수행한 연구[7]에서는 가상현실 공간 내에서 360도 환경 전체를



그림 2. 가상 환경 내 운영체제 구축 사례[7]

Fig. 2. An operating system implemented in virtual reality [7]

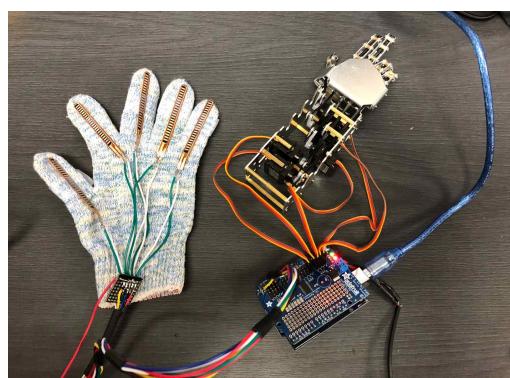


그림 3. 제안하는 프로토타입의 하드웨어 개요.

왼쪽: Flex 센서가 부착된 장갑

오른쪽 위: 손동작을 물리적으로 재현하기 위한 로봇 팔
오른쪽 아래: 3ds MAX와의 통신 수행, 로봇 팔 제어,
센서 판독을 수행하는 Arduino 및 모터 구동 회로

Fig. 3. The hardware part of the proposed prototype.

left: glove equipped with flex sensors

upper right: robot arm to replay captured figure movements

lower right: Arduino and motor shield to communicate with 3ds MAX, control the robot arm, and read flex sensors in the glove

컴퓨터 작업 공간으로 사용할 수 있는 가상 운영체제(OS) 환경을 제안하였다. 이 연구에서는 게임 엔진인 Unity3D에서 스크린 캐스터, 스크린 매니저와 파일 시스템 등을 간단하게 구현하고 기본적인 애플리케이션(메모장, 미디어 플레이어, 파일 탐색기, 캘린더, 휴지통 등)을 구현하였다. 또한 가상현실 헤드셋 기기 전면에 손가락 움직임을 인식할 수 있는 적외선 기반 센서인 LeapMotion를 부착하여 사용자가 가상공간 내에서 클릭, 선택, 가상 키보드 타이핑 등을 수행할 수 있도록 하였다(그림 2).

그러나 이 연구에서도 LeapMotion의 한계로 인해 오른손, 왼손의 인식 오류, 손가락 떨림, 특정 조명 조건 하 인식 불가능 등의 문제가 발생하였다.

본 논문에서는 앞서 논의한 문제의 해결 방안으로 손가락 동작 인식을 위한 flex 센서 기반의 저렴한 프로토타입 하드웨어

어를 구현하고 3차원 애니메이션 제작 소프트웨어(3ds MAX)에서 동작을 녹화하며 그 녹화된 결과를 3차원 모델의 애니메

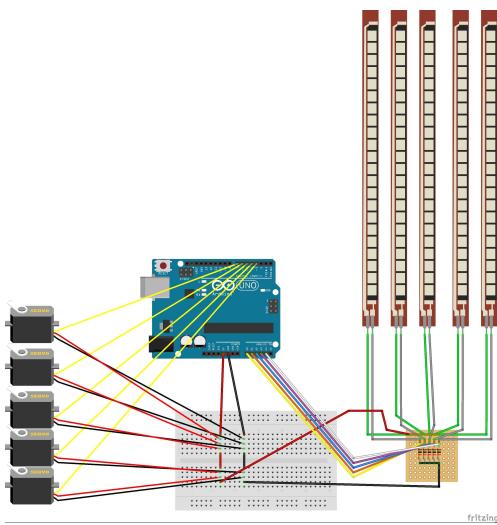


그림 4. 하드웨어 개요도

Fig. 4. Hardware schematic diagram

이션과 서보 모터 기반으로 제작된 실제 로봇 손으로 재현하는 시스템을 제안한다. 휘어지는 정도에 따라 저항 값이 변하는 flex 센서를 이용해서 손가락 움직임을 포착하였고 그 결과는 Arduino로 전송된 후 다시 PC로 전송되어 3ds MAX와 통신을 수행한다. 이렇게 전송된 손가락 관절의 기하 정보는 3ds MAX에서 MAXScript로 구현한 프로그램을 통해 keyframe[8]을 형성하고 중간 프레임이 보간으로 계산된 이후 3ds MAX 내의 손 모습의 mesh로 애니메이션을 구현한다. 또한 이렇게 포착, 재구성된 기하 정보는 서보 모터로 제작된 소형 로봇으로 전달되어 손가락의 움직임을 재현할 수 있다. 현재 구현된 프로토콜에서는 센서 입력과 로봇 연결이 로컬 범위에서 유선으로 형성되지만 본 연구에서 제안하는 동작 정보를 무선 네트워크에서도 전달할 수 있기 때문에 원격지에서의 응용(예를 들어 원격로봇 제어를 통한 동작 재현, 원격 수화 통신 또는 원격 진료)으로도 확장 가능하다. 그림 4에서는 제안하는 프로토타입 하드웨어의 전체적인 모습을 제시한다.

1-2 MAXScript

3차원 그래픽스 모델링, 렌더링, 애니메이션용 애플리케이션으로 널리 사용되는 3ds MAX는 여러 개의 독립적인 binary 모듈들(주로 dll 파일 형태)의 수명 주기를 관리하는 시스템으로도 볼 수 있다. 다시 말해 3ds MAX는 Windows 환경에서 컴포넌트 단위로 구현된 세부적인 렌더링(예. mental ray renderer), modifier (예. taper modifier 등)의 메모리 관리, 실행 환경 제공, 다른 모듈과의 상호 작용 등을 위한 일종의 운영 체제로도 볼 수 있다. 사용자는 MAX SDK[9]를 이용해서 C++ 프로그래밍 언어로 dll 형식의 새로운 모듈을 구현할 수도 있다.

3ds MAX는 직접 모듈을 구현하지 않더라도 이미 개발된 모듈들의 운영을 스크립트 프로그래밍을 통해 디자이너의 단순 반복 수작업을 자동화할 수 있는 방법을 제공한다.

예를 들어, 게임이나 영화에 적용할 대규모 3차원 배경을 제작하는 작업에서 수 천 개에 달하는 3차원 모델(예. 나무 등)을 배치해야 할 경우 많은 시간과 노동력이 투입되어야 하는 수작업을 대신해서 MAXScript[9, 10, 11]를 작성함으로써 단시간에 작업을 완료할 수 있다.

실제로 3ds MAX에서의 모든 UI 인터랙션 작업(마우스를 이용한 버튼 클릭, 모델링, 애니메이션, 렌더링 작업)은 내부적으로 동등한 MAXScript로 표현할 수 있다. 특히 3ds MAX에서는 수작업 시퀀스를 MAXScript로 변환하는 Macro Recorder라는 기능을 제공한다. 그림 5에서는 $40 \times 40 \times 30$ 크기로 높이 방향으로 5개 마디를 가지는 상자(box)를 원점(0,0,0)에 그리는 수작업을 Macro Recorder로 생성한 MAXScript 코드와 그 결과를 제시한다.

```
Box lengthsegs:1 widthsegs:1 heightsegs:5
length:40 width:40 height:30 mapcoords:on
pos:[0,0,0] isSelected:on
actionMan.executeAction 0 "40471"
```

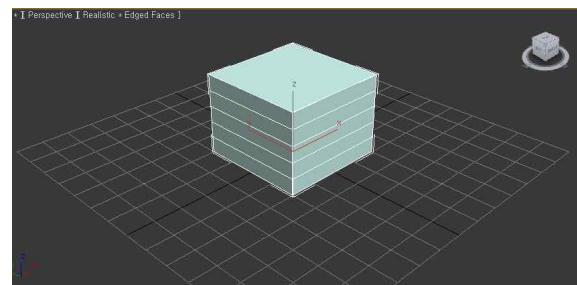


그림 5. Macro Recorder를 이용해서 생성한 MAXScript 코드와 그 결과물

Fig. 5. MAXScript code generated using Macro Recorder and its result

본 연구에서는 MAXScript를 이용해서 Arduino와의 통신을 통해 저항 값을 수신하고 그 결과를 3차원 mesh 모델과 bone에 반영한 후 이 이산적인 데이터를 통해 key frame을 적용하는 코드를 작성하였다.

II. 본 론

2-1 하드웨어 구현

1) 손가락 동작 감지를 위한 센서 회로

제안하는 프로토타입의 Arduino의 아날로그 입력 단에는 각 손가락에 부착된 flex 센서 5개를 통해 휘어짐에 비례해서 출력되는 아날로그 전압이 연결되고 디지털 출력 단은 서보 모터

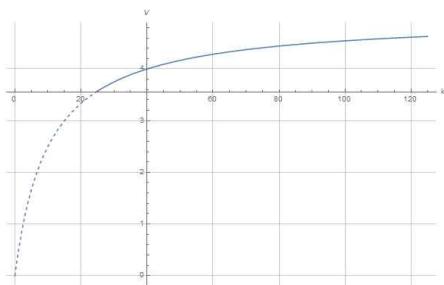
구동을 위한 전력 제공 회로와 연결되어 원하는 로봇 동작을 연출하기 위해 서보 모터의 각도를 결정한다(그림 4).

본 연구에 사용한 2.2인치 flex 센서는 휘어진 정도에 따라 다른 저항 값을 가지는 아날로그 센서이다. 이 센서는 휘어짐 정도에 따라 최소 $25\text{k}\Omega$, 최대 $125\text{k}\Omega$ 사이의 저항 값을 가지며, 5V를 $10\text{k}\Omega$ 저항과 분할하는 voltage divider 회로를 구성해서 그 분할 값을 Arduino의 아날로그 센서로 입력된다. flex 센서의 가변 저항 값을 r 이라고 하면 이 voltage divider 회로로 결정되어 Arduino로 입력되는 전압 v_i 의 범위는 다음과 같이 계산 가능하다.

$$v_i = 5 \times \frac{r}{10+r} (\text{V})$$

$$(25 \leq r \leq 125)$$

$$\Rightarrow 3.57 \text{ V} \leq v_i \leq 4.62 \text{ V}$$



위의 그래프에서 볼 수 있듯이 $[25\text{k}\Omega, 125\text{k}\Omega]$ 구간(실선 구간)에서 결정되는 전압은 저항 값과 선형적으로 비례하지 않는 특성을 가진다. 이렇게 입력된 아날로그 전압 값은 Arduino 내부에서 0부터 1023 사이의 디지털 값으로 변환되는데, 앞서 논의한 전압과 저항 값 사이의 비선형적인 특성과 함께, 사람의 손은 계속적으로 미세하게 움직이므로 사람의 손에 부착해서 구부러짐 값을 검출할 때 사람의 손이 완전하게 평평하지 않으므로 어느 정도 기본적인 휘어짐이 발생하는 문제로 인해 소프트웨어 상의 조정 과정이 필요할 수도 있으나 본 연구에서 제안하는 방식에서는 이러한 문제가 고려되지 않았다.

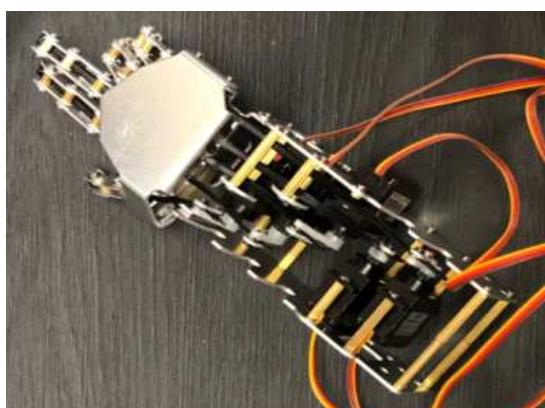


그림 6. 캡처한 움직임을 재현하기 위한 로봇팔

Fig. 6. Robot arm to replay captured motions

2) 서보 모터 기반 로봇 손가락 구현

서보 모터는 한 손가락 당 1개씩 총 5개로 구성되어 있으며 끝부분을 줄로 당겨 손가락 관절의 각도를 결정하는 간단한 방식이다(그림 6). 비록 간단한 방식이지만 end-point의 위치에 따라 관절들의 각도가 결정되는 역운동학의 물리적 구현으로 볼 수도 있다.

로봇 손가락의 끝부분을 당기기 위한 서보 모터의 회전 방향이 엄지, 검지, 중지 부분과 약지, 소지 부분의 서보 모터의 방향에 따라 Flex 센서의 값을 서보 모터 회전 값으로 적절하게 사상해야 하며 Arduino에서 다음 코드처럼 map 함수[a]를 이용해서 각도로 전환한다.

```
angle[FINGER_INDEX]=
map(finger[FINGER_INDEX],
    MIN_VOLTAGE, MAX_VOLTAGE,
    START_ANGLE, END_ANGLE);
```

2-2 소프트웨어 구현

1) 3D mesh 및 bone 모델 구조

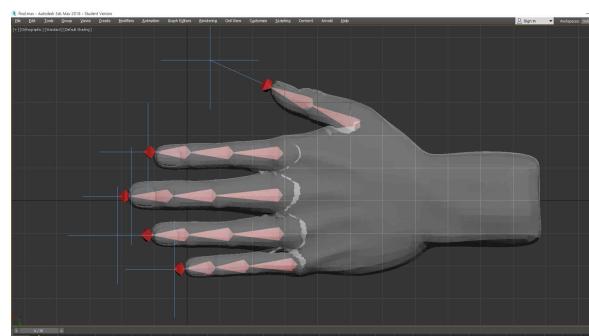


그림 7. 3ds MAX에서 제작된 손을 위한 3D mesh skin 및 bone 구조

Fig. 7. 3D mesh skin and bone structure for hand in created 3ds MAX

제안하는 프로토타입의 테스트를 위해 3ds MAX에서 제작한 3차원 손 mesh 모델에 bone을 심고 피부에 해당하는 mesh와 뼈에 해당하는 bone 구조 사이에 각 부분별로 가중치를 부여해서 연결하는 skinning 작업을 수행해서 손 모델이 bone의 움직임에 따라 함께 움직일 수 있도록 하였다(그림 7).

서보 모터의 회전 각도와 3ds MAX에서 손 애니메이션을 구현하기 위해 적용한 역운동학은 관절 끝부분의 최종 이동 지점(손가락의 경우 손가락 끝부분)인 end point의 공간상의 위치를 사용자가 지정하면 각 관절들의 각도를 컴퓨터가 계산해서 자연스러운 모습을 연출하는 방법이다. 반대로, 사용자가 각 관절들의 각도를 지정해서 end point의 위치를 결정하는 방법은 forward kinematics라고 한다. 3ds MAX에서는 메뉴 항

목 중 [Animation]-[IK Solvers]-[IK Limb Solver]를 통해 역운동학 애니메이션을 적용할 수 있다. 본 연구에서는 이 항목에서 제공하는 IK Chain을 생성하고 제어함으로써 손가락 애니메이션을 센서 입력 값과 연동하였다.

2) 3ds MAX와 Arduino의 통신

본 연구에서는 Arduino에 연결된 flex 센서의 값을 3ds MAX로 전달하기 위해 Keith Sillin의 시리얼 통신 라이브러리 [12]을 응용해서 데이터 송수신을 구현하였다. 기본적으로 시리얼 통신에서는 송수신 시에 직렬로 8 비트를 순차적으로 전송한다. 이러한 통신을 시작하기 위해서는 Arduino 측 코드에서는 `Serial.begin()` 함수를 이용해서 시리얼 통신을 초기화해야 한다. 이 함수의 인자로 baud rate를 전달하는데 이 값은 MAXScript 코드와 Arduino 코드에서 일치해야 한다. Windows 환경에서 구현된 3ds MAX 애플리케이션의 특성상 MAXScript 코드에서는 .NET [13] 객체를 사용해서 통신을 수행하였다. MAXScript에서 .NET 객체로 시리얼 통신을 수행하기 위한 객체는 다음과 같이 정의한다.

```
local serial = dotNetObject
"System.IO.Ports.SerialPort" "COM3" 9600
```

Arduino 측 코드에서 `Serial.Open()`으로 시리얼 포트를 열고 Arduino에 연결된 flex 센서의 값을 3ds MAX에서 전달받았을 때, 연속적으로 IK Chain의 정보를 변화시킬 수 있도록 0.001초마다 만료되는 타이머를 설정하였으며 `Serial.Write()`와 `Serial.Read()`를 사용해 데이터를 송수신한다. Arduino에서 flex 센서 값을 전달하면 3ds MAX 측의 MAXScript 코드에서는 전달된 데이터 값을 받아 IK Chain의 Position 속성 값을 변화시킨다. IK Chain의 위치는 손 모델의 위치, 크기, 회전에 따라 달라지기 때문에 flex 센서 값을 직접 적용할 수 없으며 각 IK Chain Position은 손가락의 처음 위치와 구부러졌을 때의 위치를 고려하여 시행착오를 통해 얻은 수치로 설정하였다. 테스트 코드에서 IK Chain position을 변경하기 위한 유사 코드는 다음과 같이 구현하였다.

```
for each finger:
    finger_segment_chain.position
        = inputDataVector - adjustedVector;
```

3) Animation Recording 기능 구현

3ds MAX의 캐릭터 애니메이션 제작 시스템의 입장에서는 지금까지 서술한 내용으로 구성되어 수신된 정보는 일정한 타

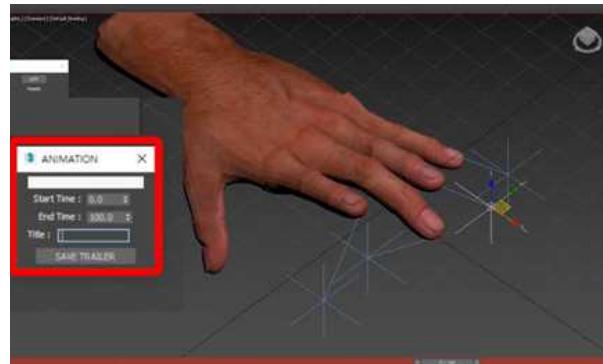


그림 8. 3ds MAX에서 구현된 프로토타입의 모습

Fig. 8. Screen shot for implemented prototype in 3ds MAX

임스탬프에 입력된 각 손가락 chain에 대한 이산적인 transform 정보이다. 이러한 이산적인 정보는 스마트 글로브를 사용하는 각 개인의 특성에 따라 차이가 있을 수도 있으며 향후 특정 애플리케이션(예. 원격 수화 애플리케이션 등)에서 인공지능 기술에 적용하기 위한 대규모 학습 데이터로 데이터베이스 등에 저장할 수 있다.

시간 축면에서 이산적인 이러한 데이터를 본 연구에서의 일차적인 목표인 3차원 애니메이션으로 전환시키기 위해서 MAXScript로 캡처한 데이터를 이용해서 key frame을 생성하고 key frame 사이에서 원하는 시간 해상도(frames per second) 만큼 frame을 생성하도록 구현하였다. 또한 디자이너가 편리하게 후처리 작업(post processing)을 수행할 수 있도록 하기 위해 3ds MAX 상에서 키보드로 Time Slider를 이동시키고 원하는 위치에서 추가적으로 key frame을 생성할 수 있도록 구현하였다.

구현한 MAXScript 파일을 3ds MAX 환경에서 실행하면 스마트 글로브의 움직임을 녹화할 수 있는 상태로 전환(3dsMAX에서 `setKeyMode == true`)되며 3ds MAX 내에 동시에 진행 중인 작업을 잠시 중단하고 기록을 위한 스크립트로 포커스를 이동시킨다. 이 스크립트로 생성된 UI 창(그림 8의 빨간색 사각형)을 종료하면 자동으로 녹화를 종료하도록 구현하였다. 이러한 작동은 3ds MAX 내 UI 창의 시작과 종료 시에 호출되는 callback 함수(`on window open do` 및 `on window close do`)를 정의해서 구현하였다. 이 두 함수의 유사 코드는 다음과 같다.

```

on window open do:
    if keyframe mode == off then:
        keyframe mode = on;
        get input focus;

on window close do:
    keyframe mode = off;

```

또한 디자이너의 작업 편의를 위해 “a”, “d”, “Space Bar”, “q”, “r”키를 사용하여 Time Slider를 움직이고 key frame을 생성, 재생, 삭제할 수 있도록 구현하였다. 실제 작동 가능한 구현된 MAXScript 코드의 일부를 다음 표에서 소개한다.

```

-- move time slider
if keyCode == virtKeys.a then (
    sliderTime -= 1f
)
if keyCode == virtKeys.d then (
    sliderTime += 1f
)
-- create key frame
if keyCode == virtKeys.space then (
    addnewkey $Chain001.controller sliderTime
    :
)
-- play animation
if keyCode == virtKeys.q then (
    max time play
)
-- delete key frame
if keyCode == virtKeys.r then (
    conP = $Chain001.controller
    num = sliderTime as integer
    selectkeys conP (num/160)
    deletekeys $Chain001 #selection
    :
)

```

구현한 MAXScript 코드에서는 수신된 센서 값을 이용해서 key frame 애니메이션만 구현하는 것에 그치지 않고 이 정보를 활용해서 다시 Arduino에 연결된 로봇팔의 손가락을 움직일 수 있다. 현재는 Arduino 한 개에 센서와 모터 구동을 위한 회로를 경유한 로봇 팔이 모두 함께 연결되어 있으나 인터넷통신 환경에서 원격으로 로봇 팔을 직접적으로 작동시키는 애플리케이션으로 쉽게 확장 가능하다고 판단한다.

III. 결 론

3-1 결과 논의

본 연구에서는 Flex 센서를 이용해 손가락 관절의 회전 각도를 감지하여 Arduino를 통해 3ds MAX로 전달한 후 이 정보를 바탕으로 3D 모델의 key frame 애니메이션을 제작하는 동시에 적절한 피드백을 위해 로봇으로 손의 움직임을 재현하였다. 특히, 서두에서 언급했던 애니메이터들이 수작업으로 미세한 부분들의 관절을 움직여 한 프레임씩 애니메이션을 작업해야 한다는 불편을 해소하기 위해 3D 모델 애니메이션을 녹화할 수 있는 기능을 구현함으로써 기존 애니메이션 제작 방식보다 쉽고 빠른 방식을 제안하였다. 손 모델, 정확하게는 손가락이 5개인 인간과 기타 의인화된 동물을 대상으로 하고 있기 때문에 형태가 다른 캐릭터(예를 들어 손가락이 6개 이상인 로봇 등)에 수정 없이 바로 적용할 경우 한계가 있을 수도 있지만, 제안한 프로토타입을 비교적 간단하게 확장해서 충분히 적용 가능하다고 판단한다. 3D 손 모델의 Transformation 정보를 변경하고서 flex 센서로 모델을 제어하고자 한다면 다시 IK Chain의 위치 점을 고려해서 값을 넣어줘야 하는 한계가 있었다. 절대적 값으로 위치를 결정하기 때문에 이러한 문제가 발생했으나 flex 센서 값을 상대적인 값으로 변환함으로써 손 모델의 Transformation 정보가 변경되어도 코드변경 없이 제어할 수 있었다. 또한 기존에 존재하는 고가의 장비 없이 저렴한 비용으로 하드웨어를 구성하고 3ds MAX와 Arduino의 연동 하에 사용자 손가락의 움직임을 가상공간에서 3차원 애니메이션으로 표현하고 3ds MAX 상에서 바로 로봇을 제어할 수 있는 애니메이션 제작 방법론을 논의하고 제시하였다. 2-1절에서 언급한 스마트 글로브 착용 시 손가락의 미세한 떨림과 voltage divider 회로의 특성으로 인한 입력 저항/ 출력 전압 사이의 비선형적 관계뿐만 아니라 온도 변화에 민감한 아날로그 수동 소자인 flex 센서 자체에서 발생하는 노이즈, 센서의 특성 상 손가락의 개별 관절의 각도를 측정할 수 없는 문제 등이 제안하는 방식의 한계라고 할 수 있으나 노이즈의 문제는 기본적으로 노이즈를 어느 정도 상쇄하는 보간 기술인 key frame 기술로 인해 큰 문제로 나타나지 않았다.

3-2 향후 연구 방향

본 연구에서 제안하는 프로토타입은 여러 사용자별로 특정 손 모양에 대한 다양한 데이터를 수집할 수 있다. 이러한 데이터는 향후 인공지능 등에 활용 가능한 대규모 학습 데이터로 활용할 수 있으리라 본다. 또한 보다 정밀한 데이터의 수집과 정교한 애니메이션 구현을 위해 앞서 3-1절에서 언급한 한계들 중 다관절 인식 시스템으로 확장함으로써 보다 정교하고 의미 있는 정보를 획득하는 방법으로 발전시켜 나가고자 한다.

감사의 글

본 연구는 덕성여자대학교 2017년도 교내연구비 지원에 의해 수행되었음.

참고문헌

- [1] Kenwright, *Game Inverse Kinematics: A Practical Introduction*, 1st ed. CreateSpace Independent Publishing Platform, 2013.
- [2] Motion Capture in wikipedia. Available:
https://en.wikipedia.org/wiki/Motion_capture
- [3] Wired glove in wikipedia. Available:
https://en.wikipedia.org/wiki/Wired_glove
- [4] Leap Motion official website. Available:
<https://www.leapmotion.com/>
- [5] Kinect in wikipedia. Available:
<https://en.wikipedia.org/wiki/Kinect>
- [6] T. Park, "Efficient Filtering for Depth Sensors under Infrared Light Emitting Sources", *The Journal of Digital Contents Society*, Vol. 13, No. 3, pp. 271-278, September 2012.
- [7] E. Kim, J. Kim, E. Yoo and T. Park, "Study on Virtual Reality (VR) Operating System Prototype", *Journal of Broadcast Engineering*, Vol. 22, No. 1, pp. 87-94, January 2017.
- [8] Keyframe animation in wikipedia. Available:
https://en.wikipedia.org/wiki/Key_frame
- [9] A. Bicalho, *Maxscript and the Sdk for 3d Studio Max*, 1st ed. Sybex Inc, 2000.
- [10] Autodesk, *3ds Max MAXScript Essentials*, 2nd ed. Focal Press. 2007.
- [11] Autodesk, *3ds Max 8 MAXScript Essentials*, 1st ed. Routledge. 2013.
- [12] "How To Interface An Arduino With 3D Studio Max," Available:
<https://www.youtube.com/watch?v=i0ZHc8Z8VrE>.
- [13] dotNet official website. Available:
<https://dotnet.microsoft.com/>

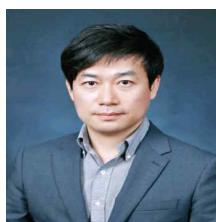
양정아(Jung Ah Yang)



2017년 : 덕성여자대학교 정보미디어대학 디지털미디어학과 학사 (공학사)
2018년 ~ 현재 : 덕성여자대학교 공과대학 IT미디어공학과 대학원 석사 과정 재학

※ 관심분야 : 컴퓨터그래픽스, 게임프로그래밍, 인공지능

박태정(Taejung Park)



1997년 : 서울대 전기공학부 (공학사)
1999년 : 서울대 전기공학부 대학원 (공학 석사, 반도체 물리 전공)
2006년 : 서울대 전기컴퓨터공학부 대학원 (공학박사, 컴퓨터 그래픽스 전공)

2006년 ~ 2013년 : 고려대학교 연구교수
2013년 ~ 2017년 : 덕성여자대학교 정보미디어대학 디지털미디어학과 조교수
2018년 ~ 현재 : 덕성여자대학교 공과대학 IT미디어공학과 부교수
※ 관심분야 : 컴퓨터그래픽스, 병렬처리, 인공지능, 수치해석, 3차원 모델링