

Design of Personalized Exercise Data Collection System based on Edge Computing

Hyon-Chel Jung*, Duk-Kyu Choi**, Myeong-Chul Park**

*Researcher, Wonju Medical Industry Techno Valley, Wonju, Korea

**Professor, Dept. of Avionics Engineering, Kyungwoon University, Gumi, Korea

**Professor, Dept. of Avionics Engineering, Kyungwoon University, Gumi, Korea

[Abstract]

In this paper, we propose an edge computing-based exercise data collection device that can be provided for exercise rehabilitation services. In the existing cloud computing method, when the number of users increases, the throughput of the data center increases, causing a lot of delay. In this paper, we design and implement a device that measures and estimates the position of keypoints of body joints for movement information collected by a 3D camera from the user's side using edge computing and transmits them to the server. This can build a seamless information collection environment without load on the cloud system. The results of this study can be utilized in a personalized rehabilitation exercise coaching system through IoT and edge computing technologies for various users who want exercise rehabilitation.

▶ **Key words:** Edge Computing, Motion Information, Pose Estimation, Rehabilitation Exercise Coaching System

[요 약]

본 논문에서는 운동 재활 서비스에 제공할 수 있는 엣지 컴퓨팅 기반의 운동 데이터 수집 디바이스를 제안한다. 기존 클라우드 컴퓨팅 방식에서는 사용자가 급증하는 경우 데이터 센터의 처리량이 증가하여 많은 지연 현상을 발생하는 문제점을 가진다. 본 논문에서는 엣지 컴퓨팅을 이용하여 사용자측에서 3차원 카메라를 통한 영상 정보를 기반으로 포즈 에스티메이션을 적용한 신체 관절의 키포인트 위치를 측정하고 추정하여 서버에 전송하는 디바이스를 설계하고 구현하였다. 본 연구의 결과를 통하여 클라우드 시스템에 부하없이 원활한 정보 수집 환경을 구축할 수 있으며 운동 재활을 원하는 다양한 사용자를 대상으로 IoT 및 엣지 컴퓨팅 기술을 통한 개인 맞춤형 재활운동 코칭 시스템에 활용될 수 있을 것이다.

▶ **주제어:** 엣지 컴퓨팅, 동작정보, 자세추정, 재활운동 코칭 시스템

-
- First Author: Hyon-Chel Jung, Corresponding Author: Myeong-Chul Park
 - *Hyon-Chel Jung (jhc@wmit.or.kr), Wonju Medical Industry Techno Valley
 - **Duk-Kyu Choi (dkchoi@ikw.ac.kr), Dept. of Avionics Engineering, Kyungwoon University
 - **Myeong-Chul Park (africa@ikw.ac.kr), Dept. of Avionics Engineering, Kyungwoon University
 - Received: 2021. 04. 20, Revised: 2021. 05. 13, Accepted: 2021. 05. 17.

I. Introduction

근래 비대면 상황이 지속되면서 재택 건강관리를 위한 다양한 요구가 제기되고 있으며 원격의료와 활성화되면서 생체정보측정과 동작을 추적하는 재활운동 솔루션 개발을 통한 서비스 시장이 커지고 있는 추세이다[1]. 생체정보 측정장치와 모션 캡처를 통한 질환별 환자 개인의 실시간 정보를 바탕으로 담당 전문가(온/오프 전문 관리자)가 적극적으로 관리하여 환자에게 최대의 병리적 호전과 건강의 증진 효과를 보일 수 있는 개인 맞춤형 운동 재활 서비스의 제공은 만성질환자들의 건강한 신체를 유지함으로써 삶의 질 향상과 의료비에 대한 부담을 해소하는데 많은 도움이 될 것이다. 이를 위하여 생애주기 개인 맞춤형 운동 재활 관리 및 처방을 위한 데이터 수집·저장·분석·서비스 플랫폼, 체형에 따른 운동 테스트 레벨 측정 및 동작 분석, 신체 부위별 운동관리 및 처방 콘텐츠, 식단 관리를 위한 AI 칼로리 계산 및 맞춤 식단 제공 서비스 그리고 이러한 처방을 제대로 수행했을 때 포인트 등으로 리워드 할 수 있는 기술 개발 추진이 필요하다. 본 연구는 이를 위한 기반 연구로서 운동을 통한 재활을 원하는 다양한 개인 사용자를 대상으로 IoT 기기 및 스마트 앱 등을 통하여 다양한 건강 정보를 플랫폼으로 수집하고, 수집된 정보를 활용하여 지식기반형의 운동재활 서비스 제공을 목표로 한다. 특히, 병원에서의 운동 재활 뿐 만 아니라 가정에서의 사용자도 전문 치료사에게 맞춤형 재활 운동을 측정, 진단, 예방, 예측, 사후관리의 생애주기에 따라 관리 받는 것과 같은 효과를 얻을 수 있는 개인 맞춤형 재활운동 코칭 솔루션 기술 개발을 근간으로 한다. 개인의 신체 및 운동 데이터를 측정하고 수집하기 위하여 본 연구에서는 엣지 컴퓨팅 기술 기반의 운동 데이터 수집 디바이스 개발하고 이를 통하여 표준 동작 영상 및 신체 데이터, 근력과 심박 데이

터와 같은 운동 빅데이터 확보를 위한 시스템을 구축하고 운동재활 분야의 기능별 운동법에 대한 영상과 함께 관련 속성 정보를 개발할 수 있도록 한다. 논문의 구성은 2장에서 엣지 컴퓨팅 기술과 헬스케어 분야에서의 적용 사례를 살펴보고 3장에서 구현을 위한 운동 데이터 수집 디바이스의 설계에 관해 기술한다. 4장에서 엣지 컴퓨팅 기반의 데이터 수집 시스템의 구현과정을 설명하고 5장에서 결과에 관해 기술한다.

II. Background

1. Edge Computing

클라우드 컴퓨팅의 사용자가 급증하면서 해당 서버와 데이터 센터의 처리량이 급속히 증가하여 수집된 데이터를 분석하고 송신하는 과정에서 지연 현상이 발생하는 문제점을 가진다. 이를 해결하기 위한 목적으로 데이터 처리를 특정 서버나 데이터 센터가 아닌 단말 단계에서 처리하는 것을 엣지 컴퓨팅이라 하며, 일종의 분산 처리를 위한 환경이라 할 수 있다. Fig. 1과 같이 클라우드 컴퓨팅의 지연시간이 100ms 이상이라면 엣지 컴퓨팅은 10ms이하로 지연시간을 단축시킬 수 있다. 엣지 컴퓨팅의 플랫폼은 각 장치에서 전송된 데이터를 PF(PeraFlop), 1초 내 1,000조 번의 연산처리가 가능한 플랫폼이다. 즉, 스마트 앱이나 장치에서 데이터의 생성과 동시에 데이터를 처리하므로 클라우드 시스템에 집중화되는 부하를 감소할 수 있고 서버의 장애로 인한 치명적인 타격에도 강인한 특징을 가진다. 또한, 즉시성이 요구되는 자율주행, 항공엔진, AR/VR, 생체 인식 및 헬스케어 등의 4차 산업혁명의 IoT 핵심 기술로 각광받고 있다[2].

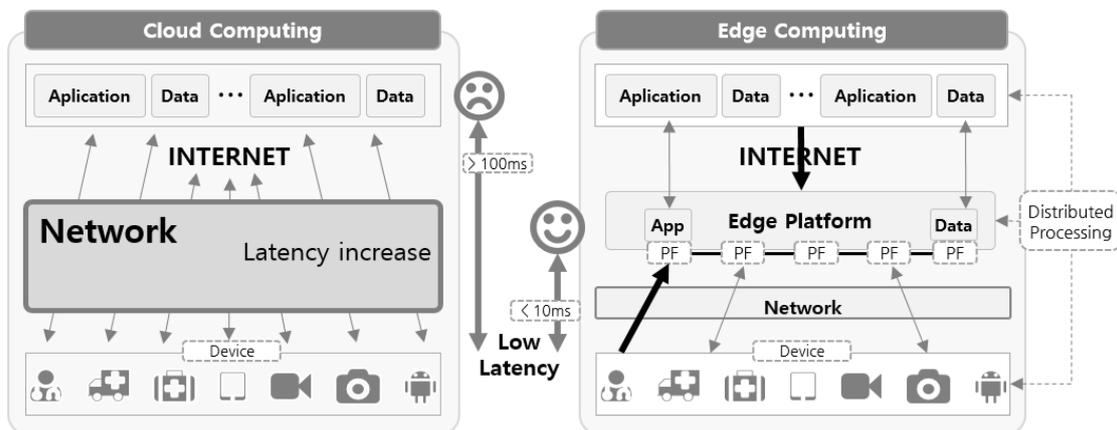


Fig. 1. Cloud computing versus Edge computing

2. Edge Computing in Healthcare

엣지 컴퓨팅이나 엣지 기반의 IoT 헬스케어 프레임워크는 스마트 센서를 통한 측정 및 진단, 예방을 위한 원격 모니터링 시스템에 대한 연구가 활발히 진행중이다[3-5]. 또한 엣지 컴퓨팅 노드를 서버로 작동하여 헬스케어 IoT 센서 데이터를 수집하고 분석 및 처리하여 신속한 대응을 위한 연구가 진행되고 있다[6]. 최근 IoT 기술의 발전으로 소프트웨어 플랫폼과 시스템 아키텍처를 활용하는 지능형 솔루션을 통하여 만성질환 모니터링, 노인 및 소아 치료, 건강 및 운동 관리 등의 헬스케어 문제를 해결하고 있다[7,8].

Chaitra Hegde[9]는 COVID-19로 인한 응급실 방문 환자수가 급증하는 추세에서 대기 시간이 감염 전파 위험에 관계성을 가진다고 가정하여 응급실의 환자를 자동으로 분류하는 라즈베리파이 기반의 오픈 소스 시스템을 제안하였다. 가시 광선과 원적외선 카메라를 통하여 열과 청색증의 두 가지 증상으로 분류하였다. Min Chen[10]는 응급 상황에 대비하고 개인화된 의료정보 서비스를 제공하기 위하여 ECC기반의 엣지 컴퓨팅을 통한 스마트 헬스케어 시스템을 소개하였으며, Soraia Oueida[11]은 병원 응급실의 복잡한 동적 행위에 대한 내부 자원 및 서비스의 흐름을 모델링 및 최적화 목적으로 맞춤형 클라우드 및 엣지 컴퓨팅과 통합된 페트리 넷(Petri net)을 사용한 RPN(Resource Preservation Net) 프레임 워크를 제안하였다. 본 논문에서는 Fig. 2와 같이 개인 맞춤형 재활운동 코칭 시스템의 동작 기반의 운동 데이터 수집을 위한 엣지 컴퓨팅 시스템을 제안하고자 한다.

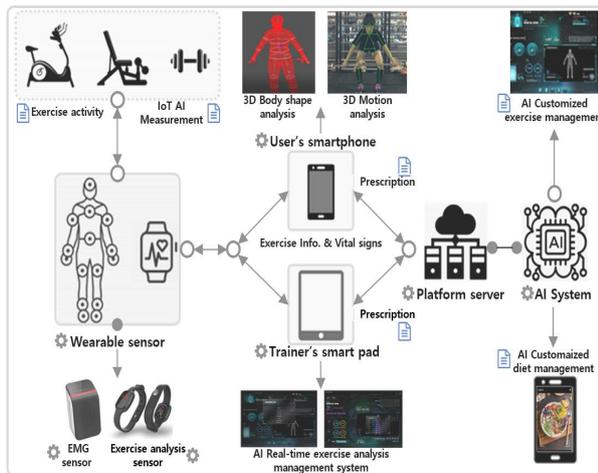


Fig. 2. Personalized Rehabilitation Exercise Coaching System

III. Design

1. Configuration of System Architecture

엣지 컴퓨팅 기반의 수집 디바이스의 전체적인 시스템 아키텍처는 Fig. 3과 같다. 엣지 디바이스(Edge Device)는 카메라의 제어와 모션캡처를 위한 연산 처리를 담당하고 포즈 에스티메이션(Pose Estimation)[12] 연산을 담당한다. 엣지 디바이스는 네트워크에 연결되지 않은 상황에서도 독립적으로 인공지능 동작이 가능한 GPU를 탑재한 임베디드 시스템으로 개발의 효율성을 높이기 위해 Nvidia-Jetpack[13]를 사용한다. 모션 캡처 영상을 WebRTC(Web Real-Time Communication)[14] 서버로 제공하기 위한 영상의 인코딩 연산 처리를 수행하며 CUDA(Compute Unified Device Architecture)[15] 기술을 사용한다. 또한, 모션 캡처 영상을 스트리밍 데이터로 웹에 제공해 주기 위해 RTP(Real-time Transport Protocol)로 WebRTC 서버에 전송한다.

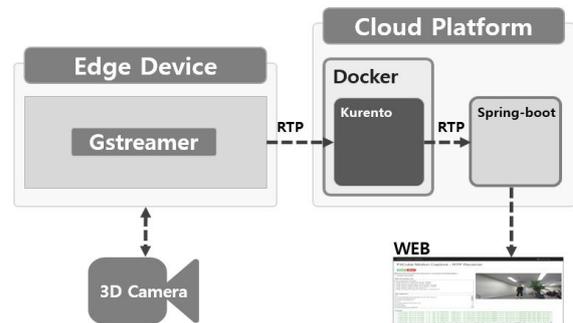


Fig. 3. Composition of system architecture

포즈 에스티메이션은 인공지능을 활용하여 사람의 신체 관절인 키포인트(Keypoint)의 위치를 측정하고 추정하는 방법이다. 연구에 적용하는 주요 키포인트는 눈, 코, 귀, 입을 포함하는 Head와 Neck, Shoulder, Elbow, Wrist, Hip, Knee, Ankle로 Head를 제외하고 각 좌우 주 부위씩 18부위를 측정한다. 모션 캡처에 오버레이하기 위한 포즈 에스티메이션 바디 트래킹(Pose Estimation Body Tracking)을 위한 구현은 C++, TensorFlow, OpenCV를 사용하고 자세한 내용은 4절에서 소개한다. 엣지 디바이스 내에는 Gstreamer를 구축하는데 이는 파이프라인(Pipeline) 기반의 스트리밍 데이터를 처리하기 위한 프레임워크이다. Gstreamer는 다수의 element로 구성되며, element는 플러그인에 의해 사용된다. 카메라 영상과 인공지능을 활용한 포즈 에스티메이션을 element로 받아 기존 Gstreamer에서 제공하는 element와 함께 파이프라인을 구성한다.

2. Configuration of Cloud Platform

클라우드 플랫폼은 쿠렌토(Kurento)[16]에서 스트리밍 데이터를 디코딩하기 위한 GPU 기반의 서버로 Fig. 2에서 언급한 코칭 서비스를 위한 웹 서버 역할을 한다. 도커(Docker)[17]는 쿠렌토 라이브러리의 효율적인 관리를 위해 사용하며 쿠렌토는 오픈소스 WebRTC 서버로 스트리밍 데이터의 관리와 처리를 담당한다. 쿠렌토의 요구사항을 충족하기 위한 서버를 별도로 준비하는 단점은 있지만 사용자의 접근성과 개발 효율성을 충족시키는 장점도 있다. 옛지 디바이스로부터 3D 카메라로 촬영한 사용자의 모션 캡처 영상을 RTP 기반의 스트리밍 데이터로 전달받아 웹에 전송하는 것이 쿠렌토의 주요 역할이 된다. 도커는 쿠렌토와 스프링부트의 라이브러리 의존성이 뒤섞이는 문제를 방지하기 위해 활용된다. 그리고 WebRTC는 웹 서비스와 GPU 연산처리를 고려하여 Naver Cloud Platform에 구축하였다. 스트리밍 데이터를 웹에 표현하기 위해 스프링부트(Spring-boot)[18]를 사용한다. 옛지 디바이스의 구성요소는 Table 1과 같다.

Table 1. Specification of Edge Device

Item	Specifications
GPU	NVIDIA Volta Architecture 384-NVIDIA CUDA Core&48-Tensor Core
CPU	6 Core NVIDIA Carmel ARM 64bit CPU 6MB L2 + 4MB L3
DLA	2 NVDLA engine
Vision Accelerator	7-Way VLIW vision processor
Memory	8GB 128bit LPDDR4x 51.2GB/s
Storage	micro SD
Video Encode	2x 4Kp30 6x 1080p 60 14x 1080p30(H.265/H.264)
Video Decode	2x 4Kp60 4x 4Kp30 12x 1080p60 32x 1080p30(H.265) etc.
Camera	2 MIPI CSI-2 PHY rane
Connectivity	Giga bit ethernet, M.2 E(WiFi/BT)
Display	HDMI and DP
USB	4 USB 3.1, USB 2.0 Micro-B
ETC	GPIO, I2C, I2S, SPI, UART
Appearance	103mm x 90.5mm x 34mm

IV. Implementation

1. Implementation of Edge Device

옛지 디바이스는 NVIDIA의 Jetson 모듈이며 AI 애플리케이션 구축의 가장 포괄적인 솔루션인 JetPack 4.4.1를 적용하였다. JetPack은 리눅스 운영체제가 포함된 드라이버 패키지와 딥 러닝, 컴퓨터 비전 및 가속 컴퓨팅과 멀티

미디어를 위한 CUDA 라이브러리와 API를 포함하고 있다. Fig. 4는 옛지 디바이스에 JetPack을 적용하는 화면이다.

```

Infracube@infracube-desktop:~$ sudo apt-cache show nvidia-jetpack
Package: nvidia-jetpack
Version: 4.4.1-b50
Architecture: arm64
Maintainer: NVIDIA Corporation
Installed-Size: 194
Depends: nvidia-cuda (= 4.4.1-b50), nvidia-openssl (= 4.4.1-b50), nvidia-cudnn8 (= 4.4.1-4.4.1-b50), nvidia-l4t-jetson-multimedia-api (>= 32.4-0), nvidia-l4t-jetson-multimedia
Homepage: http://developer.nvidia.com/jetson
Priority: standard
Section: metapackages
Filename: pool/main/n/nvidia-jetpack/nvidia-jetpack_4.4.1-b50_arm64.deb
Size: 29412
SHA256: ec502e1e3672c059d0dd49e5673c5b2d9c666504d4173ee514bb4376547a171
SHA1: 75e405f1ad533bfc04280d1f9b2378880c39be5
MD5sum: 1267b31d8b8419d9847b0ec4961b1544
Description: NVIDIA Jetpack Meta Package
Description-md5: a1462789d9bc54909ae1091d32c0a8

nvidia-jetpack
Version: 4.4.1-b144
Architecture: arm64
Maintainer: NVIDIA Corporation
Installed-Size: 195
Depends: nvidia-container-csv-cuda (= 10.2.89-1), libopencv-python (= 4.1.1-2-gd5a58aa7),
vlnfer-plugin-dev (= 7.1.0-1-cuda10.2), libvonnxparsers7 (= 7.1.0-1-cuda10.2), libnvinfer-works-tracking-dev (= 0.88.2.501), vpl-samples (= 0.2.0), tensorrt (= 7.1.0.16-1-cuda10.2),
libnvidia-container0 (= 0.9.0-beta.1), nvidia-container-csv-visionworks (= 1.6.0.501), c
amples (= 4.1.1-2-gd5a58aa75), nvidia-container-csv-cudnn (= 8.0.0.145-1-cuda10.2), pyt
dnn-doc (= 8.0.0.145-1-cuda10.2), nvidia-container-toolkit (= 1.0.1-1), libvlnfer-dev
bopenv-dev (= 4.1.1-2-gd5a58aa75), vpl-dev (= 0.2.0), vpl (= 0.2.0), libcudnn8-dev (=
licenses (= 4.1.1-2-gd5a58aa75), nvidia-container-csv-tensorrt (= 7.1.0.16-1-cuda10.2),
vidia-docker2 (= 2.2.0-1), libvisionworks-sfm (= 0.90.4.501), libnvidia-container-tools
ing (= 0.88.2.501)

```

Fig. 4. JetPack 4.4.1 Installation for Jetson Xavier NX

옛지 디바이스에는 사용자의 모션을 캡처하기 위한 카메라를 연결한다. 본 연구에 사용하는 3D 카메라는 듀얼 렌즈로 구성되며 두 개의 이미지 센서를 결합하여 공간 깊이(Depth)를 측정하는 스테레오 타입(Stereo-type) 방식을 사용한다. 측정된 깊이 데이터와 인공지능 포즈 에스티메이션 기술을 활용하여 모션 캡처 기술로 적용하였다. Table 2는 3D 카메라의 구성요소를 보인 것이다.

Table 2. Specification of 3D Camera

Item	Specifications
Output Resolution	2x (2208x1242) @ 15fps, 2x (1920x1080) @ 30fps 2x (1280x720) @ 60fps, 2x(672x376) @ 100fps
Field of View	Max. 110°(H) x 70°(V) x 120°(D)
Interface	USB 3.0/2.0
Depth Range	0.3m to 20m (1 to 65.6 ft)
Depth Accuracy	< 1% up to 3m < 5% up to 15m

포즈 에스티메이션을 위해 C++에서 TensorFlow를 활용하여 사용하는 데이터 셋인 MPII Human Pose, Leeds Sport Poses, FLIC, ITOP, DensePose-COCO, COCO 등[19-21]을 학습한다. 또한 TensorFlow의 데이터 셋 학습 성능을 높이기 위하여 NVIDIA CUDA 라이브러리를 사용하였다. 학습된 데이터 셋과 OpenCV를 활용하여 사용자 모션캡처 영상에 사람 신체의 키포인트를 기반으로 3D 스켈레톤(Skeleton)을 표현한다. Fig. 5는 모션 캡처 영상을 위한 메인 코드와 스켈레톤을 표현하기 위한 GLViewer 코드의 일부분을 보이고 있다.

```

void GLViewer::updateView(sl::Mat image, sl::Objects &objects)
{
    mtx.lock();
    image_handler.pushNewImage(image);
    BBox_obj.clear();
    objectsName.clear();
    bones.clear();
    joints.clear();
    for (auto i = objects.object_list.rbegin();
         i != objects.object_list.rend(); ++i) {
        sl::ObjectData& obj = (*i);
        if (renderObject(obj)) {
            auto clr_id = generateColorClass(obj.id);
            if (obj.keypoint.size()) {
                for (auto& limb : sl::BODY_BONES) {
                    sl::float3 kp_1 = obj.keypoint[getIdx(limb.first)];
                    sl::float3 kp_2 = obj.keypoint[getIdx(limb.second)];
                    float norm_1 = kp_1.norm();
                    float norm_2 = kp_2.norm();
                    if (std::isfinite(norm_1) && std::isfinite(norm_2)) {
                        bones.addCylinder(kp_1, kp_2, clr_id);
                    }
                }
            }
        }
    }
    ...
}

```

Fig. 5. GLViewer Function for Skeleton

이와 같이 포즈 에스티메이션을 이용한 모션캡처 수행 결과는 Fig. 6과 같다.



Fig. 6. Result of Motion capture using Pose Estimation

Gstreamer는 3D 카메라와 영상 파일의 사용자 모션 캡처 스트리밍 데이터를 RTP에 기반하여 전송할 수 있도록 인코딩하고 인공지능을 활용하여 수행한 포즈 에스티메이션의 바디 트래킹 결과 값을 영상에 오버레이하여 함께 인코딩하고 스트리밍 데이터를 WebRTC로 전송한다. Fig. 7은 WebRTC 서버에서 RTP 패킷을 수신하는 것을 보이고 있다.

```

e8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/89034065-5025-465b-9849-f92
state: DISCONNECTED, newState: CONNECTED
2020-12-23 15:53:54.252 INFO 28613 --- [ventExec-e2-t14] o.kurento.tutorial.rtp
cc74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/89034065-5025-465b-984
state: NOT_FLOWING, padName: default, mediaType: VIDEO
2020-12-23 15:53:58.994 INFO 28613 --- [ventExec-e2-t14] o.kurento.tutorial.rtp
cc74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/89034065-5025-465b-984
state: FLOWING, padName: default, mediaType: VIDEO
2020-12-23 15:59:18.459 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
cc74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/89034065-5025-465b-984
state: NOT_FLOWING, padName: default, mediaType: VIDEO
2020-12-23 15:59:19.060 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/5e56e4ff-ad9e-4c0a-81cb-
e: NOT_FLOWING, padName: default, mediaType: VIDEO
2020-12-23 15:59:20.780 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/5e56e4ff-ad9e-4c0a-81cb-
e: FLOWING, padName: default, mediaType: VIDEO
2020-12-23 15:59:27.973 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
cc74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/89034065-5025-465b-984
state: FLOWING, padName: default, mediaType: VIDEO
2020-12-23 16:00:06.489 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
cc74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/89034065-5025-465b-984
state: NOT_FLOWING, padName: default, mediaType: VIDEO
2020-12-23 16:00:07.090 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
cc74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/5e56e4ff-ad9e-4c0a-81cb-
e: FLOWING, padName: default, mediaType: VIDEO
2020-12-23 16:00:08.810 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/5e56e4ff-ad9e-4c0a-81cb-
e: FLOWING, padName: default, mediaType: VIDEO
2020-12-23 16:00:23.102 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/5e56e4ff-ad9e-4c0a-81cb-
e: NOT_FLOWING, padName: default, mediaType: VIDEO
2020-12-23 16:00:23.785 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/5e56e4ff-ad9e-4c0a-81cb-
e: FLOWING, padName: default, mediaType: VIDEO
2020-12-23 16:00:39.112 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/5e56e4ff-ad9e-4c0a-81cb-
e: NOT_FLOWING, padName: default, mediaType: VIDEO
2020-12-23 16:00:40.339 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/5e56e4ff-ad9e-4c0a-81cb-
e: FLOWING, padName: default, mediaType: VIDEO
2020-12-23 16:00:57.023 INFO 28613 --- [ventExec-e2-t15] o.kurento.tutorial.rtp
cc74ac8-6bfa-4a0a-8e40-8a37218086e9_kurento.MediaPipeline/89034065-5025-465b-984
state: FLOWING, padName: default, mediaType: VIDEO

```

Fig. 7. Receiving RTP packets from WebRTC server

Gstreamer element 사용하는 플러그인 목록은 Table 3과 같다.

Table 3. List of Plugins used by Gstreamer element

Plugin	Specifications
zedsrc	Place 3D camera information (color images, depth maps, etc.) into the Gstreamer pipeline
queue	Separation of threads between elements
zedodoverlay	3D camera information and skeleton of motion capture
autovideoconvert	Choose the right color for your video
video/x-raw, format=I420	Create raw video in I420 format
x264enc	Encode raw video to MPEG-4 AVC (Advanced Video Codec), H264
rtph264pay	H264 encoded data is stored in RTP packet
rtpsession	Session manager : receiving RTP and RTCP packets and managing sessions
udpsink	Used to transmit UDP packets over the network

2. Implementation of Cloud Platform

오픈소스 WebRTC 서버인 쿠렌토는 미디어 서버 및 클라이언트를 포함하고 있다. 옛지 디바이스로부터 3D 카메라로 촬영한 사용자의 모션 캡처 영상을 RTP 기반의 스트리밍 데이터로 전달받아 웹에 전송하는 역할을 한다. 쿠렌토는 수신 받은 스트리밍 데이터의 디코딩을 위해 Gstreamer를 사용하며 디코딩 시, GPU 기반의 CUDA 라이브러리 사용을 위해 Naver 클라우드 플랫폼 GPU 서버를

사용한다. 웹 서버로 Spring-boot를 사용하며 쿠렌토와 Spring-boot의 라이브러리 의존성이 섞이는 문제를 방지하기 위해 도커를 활용한다. 모션 캡처 영상으로부터 신체의 각 키포인트 18개 부위의 3차원 좌표를 출력하게 된다. Fig. 8은 쿠렌토 RTP 패킷을 받기 위한 애플리케이션의 자바 코드의 일부분을 보인 것이고 Fig.9는 쿠렌토와 엣지 디바이스 간의 핸들러 자바 코드의 일부분을 보인 것이다.

```
public class Application implements WebSocketConfigurer
{
    @Bean
    public Handler handler()
    {
        return new Handler();
    }
    @Bean
    public KurentoClient kurentoClient()
    {
        return KurentoClient.create();
    }
    @Bean
    public ServletServerContainerFactoryBean
    createServletServerContainerFactoryBean() {
        ServletServerContainerFactoryBean container = new
        ServletServerContainerFactoryBean();
        container.setMaxTextMessageBufferSize(32768);
        return container;
    }
    @Override
    public void
    registerWebSocketHandlers(WebSocketHandlerRegistry registry)
    {
        registry.addHandler(handler(), "/rtpreceiver");
    }
    public static void main(String[] args) throws Exception
    {
        SpringApplication.run(Application.class, args);
    }
}
```

Fig. 8. Application java code to receive Kurento RTP packet

```
public void afterConnectionClosed(final WebSocketSession
    session, CloseStatus status) throws Exception
    {
        log.debug("[Handler::afterConnectionClosed] status: {}",
            sessionId: {}, status, session.getId());
        stop(session);
    }

    @Override
    protected void handleMessage(WebSocketSession
        session, TextMessage message) throws Exception
    {
        JsonObject jsonMessage = gson.fromJson(
            message.getPayload(),JsonObject.class);
        String sessionId = session.getId();
        log.debug("[Handler::handleTextMessage] {}", sessionId:
            {}, jsonMessage, sessionId);
        try {
            String messageId = jsonMessage.get("id").getAsString();
            switch (messageId) {
                case "PROCESS_SDP_OFFER":
                    handleProcessSdpOffer(session, jsonMessage);
                    break;
                case "ADD_ICE_CANDIDATE":
                    handleAddIceCandidate(session, jsonMessage);
                    break;
                case "STOP":
                    handleStop(session, jsonMessage);
                    break;
                default:
                    sendError(session, "Invalid message, id: " +
                    messageId);
                    break;
            }
        }
    }
```

Fig. 9. Handler java code between Kurento and Edge Device

쿠렌토와 Spring-boot 연동 및 실행 결과는 Fig. 10과 같다.

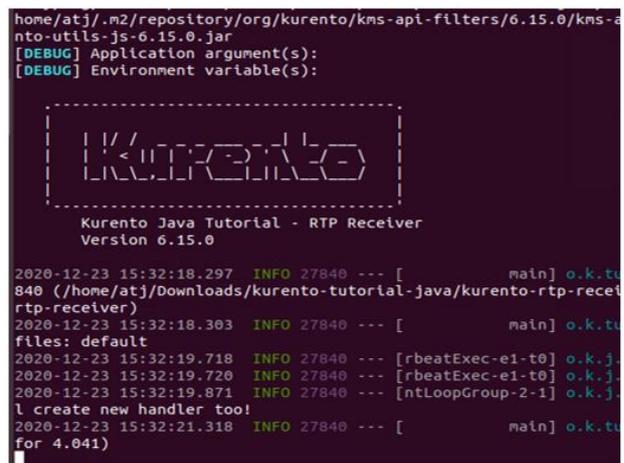


Fig. 10. Interlocking and execution of kurento and spring-boot

모션 캡처 기술 기반의 WebRTC 실행을 통한 웹의 초기 화면과 모션 캡처 스트리밍 데이터를 수신하기 위한 대기 화면은 Fig. 11과 같다.

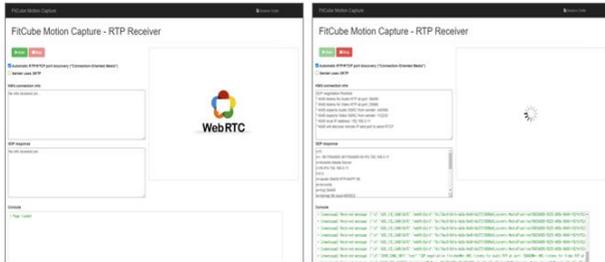


Fig. 11. Web initial screen and motion capture streaming data reception standby screen

엣지 디바이스에서 전송된 실시간 모션 캡처 영상의 웹 스트리밍 화면은 Fig. 12와 같다.

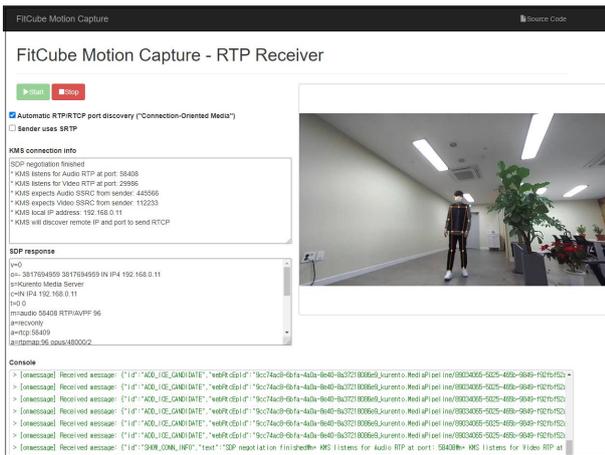


Fig. 12. Web streaming screen of motion captured video

Fig. 13은 최종 동작 정보가 3차원 좌표값으로 클라우드 서버에 전송된 결과를 보이고 있다.

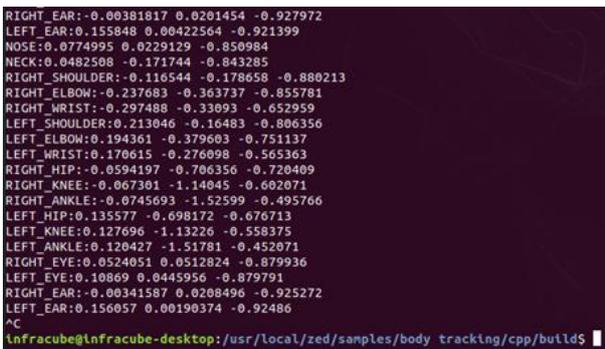


Fig. 13. Results the value of each keypoint of the body of the motion capture image

V. Conclusions

본 논문은 개인 맞춤형 재활운동 코칭 시스템에 적용할 수 있는 동작기반의 운동 데이터 수집을 위한 엣지 컴퓨팅 시스템을 제안하였다. 병원 밖의 일상생활 속에서 개인의 신체 및 운동 데이터 측정을 통하여 환자의 실시간 건강 및 체력상태를 반영하여 클라우드 시스템으로 전송하는 것은 다수의 사용자 플랫폼에서는 시간 지연등의 문제가 발생한다. 이를 극복하기 위하여 본 논문에서는 단말 단계인 사용자측에서 측정된 정보를 선처리하므로 클라우드 시스템의 부하를 줄일 수 있고 비대면 상황에 적합한 개인 맞춤형 정보 수집을 위한 디바이스 설계 및 적용을 실현하였다. 구현된 엣지 디바이스 및 WebRTC는 오픈소스 기반에서 동작할 수 있도록 설계되었으며 Spring-boot를 사용하여 웹으로 스켈레톤 기반의 모션캡처를 확인할 수 있도록 하였다. 또한 클라우드 시스템에 전송된 사용자의 동작 측정을 통한 스켈레톤의 공간 정보 값이 올바르게 산출되는 결과를 확인할 수 있었다. 기존 생체신호 측정 장치는 측정 정보만을 제공하는 기능적 제한성을 가지고 있으나, 본 연구의 결과물은 측정된 단위 영상을 포즈 에스티메이션 등의 처리를 통하여 바디 트래킹 결과값을 서버로 전송하는 기능적 특징을 가지며 요구사항 변화에 따른 단말 영역에서 각종 처리를 추가적으로 수행할 수 있는 장점을 가진다. 이를 통하여 재활 영역별 빅데이터 수집이 용이하고 체계적인 개인 맞춤형 재활 서비스제공이 가능할 것으로 사료된다. 본 연구 결과의 측정 및 수집된 동작 정보를 기반으로 실시간 운동 분석 관리 시스템으로 발전시켜 개인 맞춤형 운동 처방이나 운동관리 및 식단 관리등의 재활 코칭 시스템으로 단계적 연구를 지속할 계획이다.

REFERENCES

- [1] Maarten G. Lansberg, Catherin Legault, Adam MacLellan, Alay Parikh, Julie Muccini, Michael Mlynash, Stephanie Kemp, "Home-based Virtual Reality Therapy for Hand Recovery After Stroke," Physical medicine and rehabilitation, to be published. doi: 10.1002/pmrj.12598.
- [2] Keith Shaw, "What is edge computing and why it matters," <https://www.networkworld.com/article/3224893/what-is-edge-computing-and-how-it-s-changing-the-network.html>
- [3] A. O. Akmandor and N. K. Jha, "Smart health care: An edge-side computing perspective," IEEE Consum. Electron. Mag., vol. 7, no. 1, pp. 29-37, Jan. 2018.

- [4] M. S. Hossain and G. Muhammad, "Cloud-based collaborative media service framework for HealthCare," *Int. J. Distrib. Sensor Netw.*, Vol. 10(3), Art. no. 858712, Mar. 2014.
- [5] G. Muhammad, M. S. Hossain, and N. Kumar, "EEG-based pathology detection for home health monitoring," *IEEE J. Sel. Areas Commun.*, early access, Aug. 31, 2020. doi: 10.1109/JSAC.2020.30206
- [6] L. Liu, Z. Chang, and X. Guo, "Socially aware dynamic computation of offloading scheme for fog computing system with energy harvesting devices," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1869-1879, Jun. 2018.
- [7] M. S. Hossain and G. Muhammad, "Deep learning based pathology detection for smart connected healthcare," *IEEE Netw.*, vol. 34, no. 6, pp. 120125, Dec. 2020. DOI : 10.1109/MNET.011.2000064
- [8] Y. Abdulsalam and M. S. Hossain, "COVID-19 networking demand: An auction-based mechanism for automated selection of edge computing services," *IEEE Trans. Netw. Sci. Eng.*, early access, Sep. 24, Sep. 2020. DOI : 10.1109/TNSE.2020.3026637
- [9] C. Hegde, P. B. Suresha, J. Zelko, Z. Jiang, R. Kamaleswaran, M. A. Reyna, and G. D. Clifford, "Autotriage-an open source edge computing raspberry pi-based clinical screening system," *MedRxiv*, to be published. DOI : 10.1101/2020.04.09.20059840
- [10] Min Chen, Wei Li, Yixue Hao, Yongfeng Qian, Iztok Humar, "Edge cognitive computing based smart healthcare system," *Future Generation Computer Systems*, Vol. 86, pp. 403-411, Sep. 2018. DOI : 10.1016/j.future.2018.03.054
- [11] Oucida Soraia, Kotb Yehia, Aloqaily Moayad, Jararweh Yaser, Baker Thar, "An Edge Computing Based Smart Healthcare Framework for Resource Management," *Sensors*. Vol. 18(12): 4307, Dec. 2018. DOI : 10.3390/s18124307
- [12] Sun ke, Bin Xiao, Dong Liu, Jingdong Wang, "Deep High-Resolution Representation Learning for Human Pose Estimation," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5686-5696, Feb. 2019. DOI : 10.1109/CVPR.2019.00584
- [13] JetPack SDK, <https://developer.nvidia.com/embedded/jetpack>
- [14] WebRTC, <https://webrtc.github.io/webrtc-org/architecture/>
- [15] Huayou Su, Mei Wen, Nan Wu, Ju Ren, and Chunyuan Zhang, "Efficient Parallel Video Processing Techniques on GPU: From Framework to Implementation," *The Scientific World Journal*, Volume 2014, Article ID 716020, Mar 2014. DOI : 10.1155/2014/716020
- [16] Kurento, <https://www.kurento.org/documentation>
- [17] R. Y. Jang, R. Lee, M. W. Park, and S. H. Lee, "Development of an AI analysis service system based on OpenFaaS," *J. KCA*, Vol. 20(7), pp. 97-106, July 2020. DOI : 10.7840/kics.2021.46.2.390
- [18] Spring Boot, <https://spring.io/projects/spring-boot>
- [19] MPII Human Pose, <http://human-pose.mpi-inf.mpg.de/>
- [20] Christian Rupprecht, Iro Laina, Robert DiPietro, Maximilian Baust, Federico Tombari, Gregory Hager, Nassir Navab, "Learning in an Uncertain World: Representing Ambiguity Through Multiple Hypotheses," *IEEE International Conference on Computer Vision 2017*, Aug. 2017. DOI : 10.1109/ICCV.2017.388
- [21] R. A. Güler, N. Neverova and I. Kokkinos, "DensePose: Dense Human Pose Estimation in the Wild," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7297-7306, Feb. 2018. DOI : 10.1109/CVPR.2018.00762.

Authors



Hyon-Chel Jung received B.S degree in Biomedical Engineering Konkuk University in 2014, a M.S degrees in Biomedical Engineering from Konkuk University in 2017. PhD Candidate in Health and Environmental

Science from Korea University in 2019. He is currently a Researcher in Wonju Midecal Industry Techno valley. He is interested in digital convergence business model, Health Cloud, mobile Health Care Service.



Duk-Kyu Choi received a B.S. degree in Electronic Engineering from Kyungpook National University in 1990, a M.S. and Ph.D. degrees in Electronic Engineering from Kyungpook National University in 1993 and

1997, respectively. He is currently a Professor in the Department of Avionics Engineering, KyungWoon University. He is interested in Embedded system, Digital video technology, Mobile Telecommunication.



Myeong-Chul Park received a B.S. degree in Computer Science from Korea National Open University in 1999, and the M.S. and Ph.D. degrees in Computer Science from GyeongSang National University in 2002 and

2007, respectively. He is currently a Professor in the Department of Avionics Engineering, KyungWoon University. He is interested in Visualization, Simulation, Education of Software, Virtual Reality, and Parallel Programming.