Technical note

Remote motion capture protocol for transferring 3D motion data to remote locations in real time

Shun IRIE[1,†,*], Takashi KOBORI[1†], Kazuki SUMI[1†], Atsumichi TACHIBANA[2], Yihan QIAN[3], and Kaori MAKIFUJI[3]

1 Division for Smart Healthcare Research, Dokkyo Medical University, Mibu, Tochigi, Japan
2 Department of Anatomy, Dokkyo Medical University, Mibu, Tochigi, Japan
3 e-skin Business division, Xenoma Inc., Tokyo, Japan
[†] Equally contributed to this work
[*] Correspondence: s-irie360@dokkyomed.ac.jp; Tel.: +81-282-87-2492

Running Title: Real-time remote 3D motion data transfer protocol

Abstract

The demand for telemedicine systems has increased following the COVID-19 pandemic. Conventional telemedicine systems provide video and acoustic transferring functions using smartphones and other communication systems. However, it is difficult to use conventional telemedicine systems in rehabilitation fields because they require real-time motion data from patients. Here, we suggest a novel system for transferring three-dimensional (3D) motion data to remote places in real time, which is named the “remote motion capture protocol (RMCP).” The RMCP runs on the conventional game engine and multiple operation systems, such as Windows, Mac OS, and Android. It can transfer the 3D motion data of users to remote locations with a delay of less than 500 ms using remote procedure calling. Additionally, any type of motion capture is available in this system if it is compatible with the conventional game engine. This technical note describes the basis of the system, sample applications, and usages of RMCP.

Keywords: remote motion capture, 3D motion data, telemedicine

1. Introduction

In recent years, the demand for telemedicine systems has surged, largely driven by the challenges posed by the COVID-19 pandemic. The concept of telemedicine has existed since 1924, but the first operational system was launched in 1977 [1]. This pioneering telemedicine system comprised mobile camera units and a monitoring center for remote users and medical centers, respectively [2,3]. The remote users, which included physicians and nurses along with patients, received real-time treatment instructions from the medical center. Although a video-based system was too expensive at the time, the developed system was recognized for its advantages in communication quality over auditory-based systems, particularly in clinical settings.

Since 2010, rapid advancements in smartphone technologies have contributed to popularizing telemedicine systems as they can be used to easily transmit videos of patients and health-related data [1,4]. Recent technological developments have enabled the use of telemedicine systems in home medicine, going beyond their role as hubs between primary care physicians and medical centers [5–7].

In particular, the field of rehabilitation has experienced significant advancements through the application of telemedicine techniques known as telerehabilitation [8]. Traditionally, most telerehabilitation systems have incorporated video and auditory communication systems, alongside conventional telemedicine systems [9,10]. In recent developments, telerehabilitation has evolved to include real-time transmission of vital motion data via the Internet. It opens up possibilities for replicating realistic rehabilitation environments in cyber spaces, such as the metaverse, using virtual reality techniques [11,12]. However, existing telerehabilitation techniques, particularly those involving the transmission of motion data, often lack compatibility across various platforms and devices.

Therefore, in this study, we introduce a novel three-dimensional (3D) motion transportation technique termed “remote motion capture protocol (RMCP)” in the Unity C# environment, which enables developers of telemedicine to include the motion capture system in their own applications. This technical note describes the system overview and experimental conditions, and provides a description of each script and a sample application.

2. Systems

Figure 1 presents an overview of the systems employed in this study. The RMCP extracts bone information (root position and local joint rotations) from avatar objects controlled by a motion capture system at the patient-end. The extracted bone information is transferred to the doctor-end in real time. Furthermore, the avatar object can be moved in the same manner as the patient using remote procedure call (RPC) functions.

In Unity environments, the RMCP is independent of the motion capture system. Furthermore, the scripts for the motion capture system in the patient-end were disabled in the doctor-end to prevent conflict errors. In this study, three types of motion capture systems are used, and a depth camera and inertial measurement unit (IMU) sensors are used to create test environments.

2-1. Motion-capture system

For motion capture, we employed an IMU-based system (e-skin MEVA; Xenoma Inc., Tokyo, Japan) and depth-imaging-based system (iPi Studio; iPi Soft LLC, Moscow, Russia) for verification. The e-skin MEVA utilizes IMU sensors embedded in wearable e-textile clothing [13]. For the depth-based system, the depth camera (Azure Kinect DK, Microsoft, Redmond, WA) was located 3.0 m from a test user, with specific camera parameters (Height: 1.1 m, accelerometer, x: $1.78\ \mathrm{m/s^2}$, y: $0.01\ \mathrm{m/s^2}$, and z: $-9.72\ \mathrm{m/s^2}$). For each motion capture system, we utilized a sample scene from official Unity SDKs for further development. The proposed RMCP is available for any motion capture system with Unity C# SDKs.

2-2. Server system

We used a photon server (Exit Games, Hamburg, Germany) with photon unity networking 2 (PUN2). As shown in Fig. 1, the system environment is divided into two ends: patient (motion capture) and doctor (remote receiver). At both ends, the prefab named “ServerPrefab.prefab” with C# scripts (ServerPrefab.cs) was placed on the hierarchy. This prefab has the functions of setting up the server and instantiates the avatar when the system is switched on (Fig. 2A). It is necessary to sign up for the photon server account to obtain the app ID from the photon dashboard.

In addition, we created another C# script named “RMCprotocol.cs” and attached it with scripts of the motion capture system onto the avatar prefab in the resource folder. The type of avatar must be set as humanoid in Unity environments. The script loads the avatar information when it is instantiated, and the bone information comprising the local rotation of each bone and root position is continuously transferred as byte array data (Fig. 2B). The conversion of bone data to a byte array was conducted only when the avatar was at the patient-end. The motion of the avatar instantiated at the beginning of the patient-end was updated at the doctor-end using the RPC of the photon server, thereby enabling the simultaneous display of the avatar motion at both ends. Additional information on the application programming interface (API) can be found in API.md.

2-3. Sample applications

Figure 3 shows two personal computer displays of sample applications with the RMCP for e-skin MEVA (A) and iPi Studio with Azure Kinect DK (B). The displays were merged using an HDMI capture system (USB-CVHDUVC5; Sanwa Supply, Okayama, Japan). The sample applications comprised patient- and doctor-end systems connected to another Wi-Fi network (5 GHz bands). Test users in our team were asked to perform the two poses shown in Fig. 3C.

Subsequently, we compared motions of the avatar presented on the two computers, simultaneously. Evidently, the avatar motions in the patient-ends were successfully transported to the doctor-ends within a delay of ~500 ms in both the conditions using e-skin MEVA and Azure Kinect DK (Figs. 3A–B).

2-4. Usage note

We verified that OS such as Windows (beyond version 10), Mac OS (beyond version 13.4.1), and Android (beyond version 13) can be used with this system. The development deployment must be turned off while building in Android environments. All permissions for copyrights belong to Dokkyo Medical University; however, the source codes of this system are available to all users without any confirmation if the citation information is provided in the codes and manuscripts of the research.

2-5. Code availability

The source code can be downloaded from the GitHub repository (https://github.com/shun-irie/RMCP_Dokkyo) following the publication of this technical note. Moreover, sample applications (source code and/or compiled applications) are available upon request (https://forms.gle/AY6vgDyH7oqRAftw5) from the corresponding author.

3. Discussion

We demonstrated the use of the RMCP for IMU-based and depth-camera-based motion capture systems. In the RMCP, anyone can transfer 3D motion capture data to a remote location in real time if the motion-capture system has SDKs for Unity environments. In our survey, most motion-capture systems employed Unity SDKs as developers (Table 1). Systems and platforms that can transfer the motion data virtual motion capture (VMC) protocol Marionette (https://protocol.vmc.info/), VRChat (https://hello.vrchat.com/), and NeosVR (https://neos.com/). However, certain motion capture systems are compatible with these systems and platforms owing to lack of common standards for motion capture in this field. On the other hand, the RMCP just requires a unity plugin in their motion capture systems, which enables the engineers to develop a system using the best devices based on their requirements, environments, and costs.

Figure 3 presents a comparison of motion capture screens between patient- and doctor-ends, indicating that the delay is within 500 ms, which is close to that of conventional video chats such as Zoom [14]. Therefore, we believe that the system is suitable for telerehabilitation if the voice communication functions are included.

However, the RMCP has several limitations. First, it is only available in Unity environments and not in the Unreal Engine. Second, a motion-capture system whose TCP and UDP ports are the same as those of the photon cloud is unavailable. In this case, the developer must construct a gaming server developed by the photon system in its own network and set the port number to avoid interference. Third, the quality of motion data visualization is based on the hardware specifications because the total file size of the software compiled by Unity environments is large. Therefore, we strongly recommend that developers use avatars with few polygons.

4. Conclusion

RMCP is a novel technique for transferring 3D motion capture data to remote locations in a Unity environment. It can be utilized for any motion capture system that satisfies the following two conditions: (1) The motion capture system has SDKs for Unity environments and (2) there is no interference at the TCP/UDP ports between the photon cloud and motion capture systems.

In this study, we demonstrated two sample use cases for IMU-based and depth-camera-based motion capture systems. However, this is only a preliminary version of the software; therefore, further development is required to apply it in the field of telerehabilitation.

Author contributions

Software development: S.I., T.K., and K.S. Conceptualization: S.I. and A.T. Experiments: S.I., K.M. and Q.Y. Writing manuscripts: S.I. Review of manuscripts: S.I., Q.Y., and A.T.

Funding

This study was supported by MEXT KAKENHI (S.I., no. 22H05220 and 24H00909) and Research Support Award 2023 of the Dokkyo International Medical Education and Research Foundation (S.I., no. R5-12), and JKA and its promotional funds from KEIRIN RACE (no. 2023M-332).

Acknowledgement

We thank Dr. Ichiro Amimori (CEO of Xenoma Inc.) and Ms. Mariko Ito (technical staff of Dokkyo Medical University) for their assistance.

Conflict of Interests

S.I. received a grant donated by Dokkyo Medical University as research funding for studies conducted on the application of wearable e-textiles in healthcare fields from Xenoma Inc. on July 31, 2021.

References

[1] Jagarapu J, Savani RC: A brief history of telemedicine and the evolution of teleneonatology. Semin Perinatol **45**(5), 151416, 2021.

[2] Grundy BL, Crawford P, Jones PK, Kiley ML, Reisman A, Pao YH, Wilkerson EL, Gravenstein JS: Telemedicine in critical care: An experiment in health care delivery. J Am

College Emergency Phys. **6**(10), 439–44, 1997.

[3] Murphy RL, Bird KT: Telediagnosis: a new community health resource. Observations on the feasibility of telediagnosis based on 1000 patient transactions. Am J Public Health. **64**(2), 113–119, 1974.

[4] MacKinnon GE, Brittain EL: Mobile health technologies in cardiopulmonary disease. Chest. **157**(3), 654–664, 2020.

[5] Palmdorf S, Stark AL, Nadolny S, Eliaß G, Karlheim C, Kreisel SH, Gruschka T, Trompetter E, Dockweiler C: Technology-assisted home care for people with dementia and their relatives: Scoping review. JMIR Aging. **4**(1), e25307, 2021.

[6] Van Den Heuvel JF, Groenhof TK, Veerbeek JH, Van Solinge WW, Lely AT, Franx A, Bekker MN: eHealth as the next-generation perinatal care: an overview of the literature. J Med Internet Res. **20**(6), e202, 2018.

[7] Farias FA, Dagostini CM, Bicca YD, Falavigna VF, Falavigna A: Remote patient monitoring: A systematic review. Telemed e-Health. **26**(5), 576–583, 2020.

[8] Agostini M, Moja L, Banzi R, Pistotti V, Tonin P, Venneri A, Turolla A: Telerehabilitation and recovery of motor function: a systematic review and meta-analysis. J Telemed Telecare. **21**(4), 202–213, 2015.

[9] Kenis-Coskun O, Imamoglu S, Karamancioglu B, Kurt K, Ozturk G, Karadag-Saygi E: Comparison of telerehabilitation versus home-based video exercise in patients with Duchenne muscular dystrophy: A single-blind randomized study. Acta Neurol Belg. **122**(5), 1269–1280, 2022.

[10] Özden F, Sarı Z, Karaman ÖN, Aydoğmuş H: The effect of video exercise-based telerehabilitation on clinical outcomes, expectation, satisfaction, and motivation in patients with chronic low back pain. Irish J Med Sci. **191**(3), 1229–1239, 2022.

[11] Berton A, Longo UG, Candela V, Fioravanti S, Giannone L, Arcangeli V, Alciati V, Berton C, Facchinetti G, Marchetti A, Schena E: Virtual reality, augmented reality, gamification, and telerehabilitation: psychological impact on orthopedic patients' rehabilitation. J Clini Med. **9**(8), 2567, 2020.

[12] Truijen S, Abdullahi A, Bijsterbosch D, van Zoest E, Conijn M, Wang Y, Struyf N, Saeys W: Effect of home-based virtual reality training and telerehabilitation on balance in individuals with Parkinson disease, multiple sclerosis, and stroke: A systematic review and meta-analysis. Neurol Sci. **43**(5), 2995–3006, 2022.

[13] Fukuoka T, Irie S, Watanabe Y, Kutsuna T, Abe A: The relationship between spatiotemporal gait parameters and cognitive function in healthy adults: protocol for a cross-sectional study. Pilot Feasibility Stud. **8**(1), 1–3, 2022.

[14] Boland JE, Fonseca P, Mermelstein I, Williamson M: Zoom disrupts the rhythm of conversation. J Exp Psychol: General. **151**(6), 1272, 2022.

Table 1. Support of the Unity SDK

System	Company	Unity SDK	RMCP availability
VICON	Vicon Motion Systems	✓	?
OptiTrack	NaturalPoint, Inc.	✓	?
Xsens	Xsens Technologies B.V.	✓	?
Perception Neuron	NOITOM LTD.	?	?
e-skin MEVA	Xenoma Inc.	✓	✓
mocopi	SONY	✓	✓
Azure Kinect DK	Microsoft	✓	✓

✓ Available ? not checked

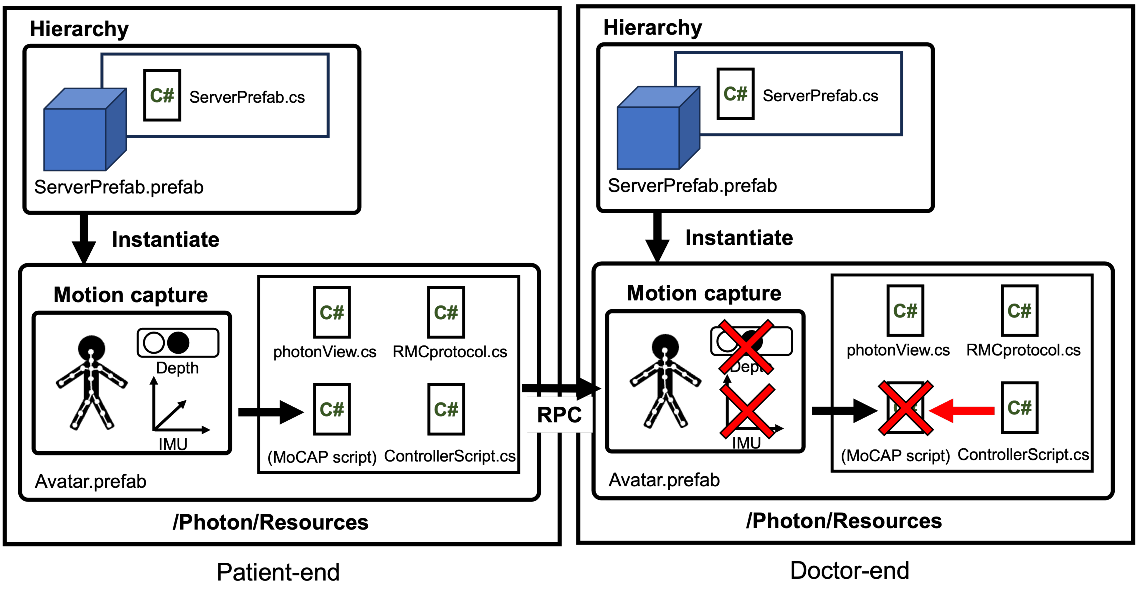

Fig. 1 Overview of RMCP

The system is divided into two main parts: the patient-end and doctor-end. ServerPrefab.cs in the hierarchy instantiates the avatar from the Resources folder. Motion capture data are collected by scripts provided by motion capture vendors (MoCAP script). RMCprotocol.cs transfers the rotation and position data as a byte array to the doctor-end using RPC, which enables avatar motion to be synchronized at both ends. At the doctor-end, ControllerScript.cs disables the MoCAP script to prevent conflict errors caused by the null reference of the motion capture system.

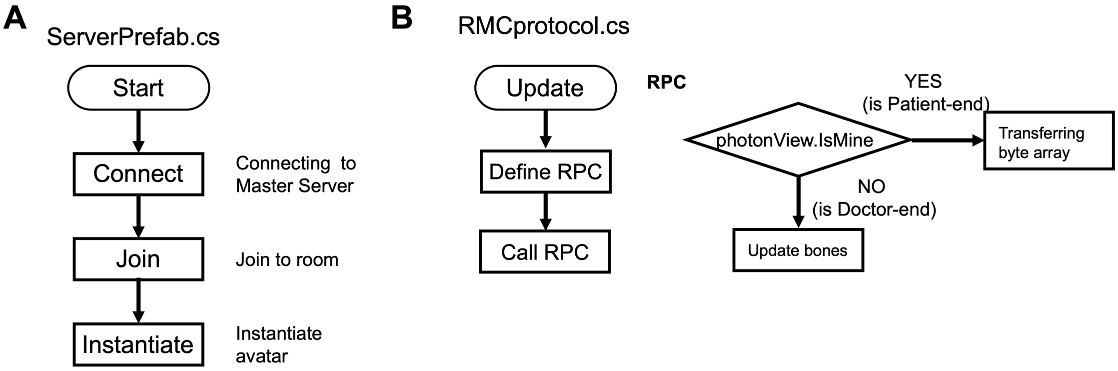

Fig. 2. Flowcharts of the source codes

A: ServerPrefab.cs. The script connects to the photon server, joins the room, and instantiates the avatar. **B**: RMCprotocol.cs. Once the avatar is instantiated, the animator of the avatar, including information regarding the bones, is set. Every 0.03 s at the patient-end, the 3D bone information of the avatar is converted to byte arrays and transferred to the doctor-end (photonView.IsMine = TRUE). In contrast, the received bone information is used to update the avatar bones at the doctor-end using RPC (photonView.IsMine = FALSE).

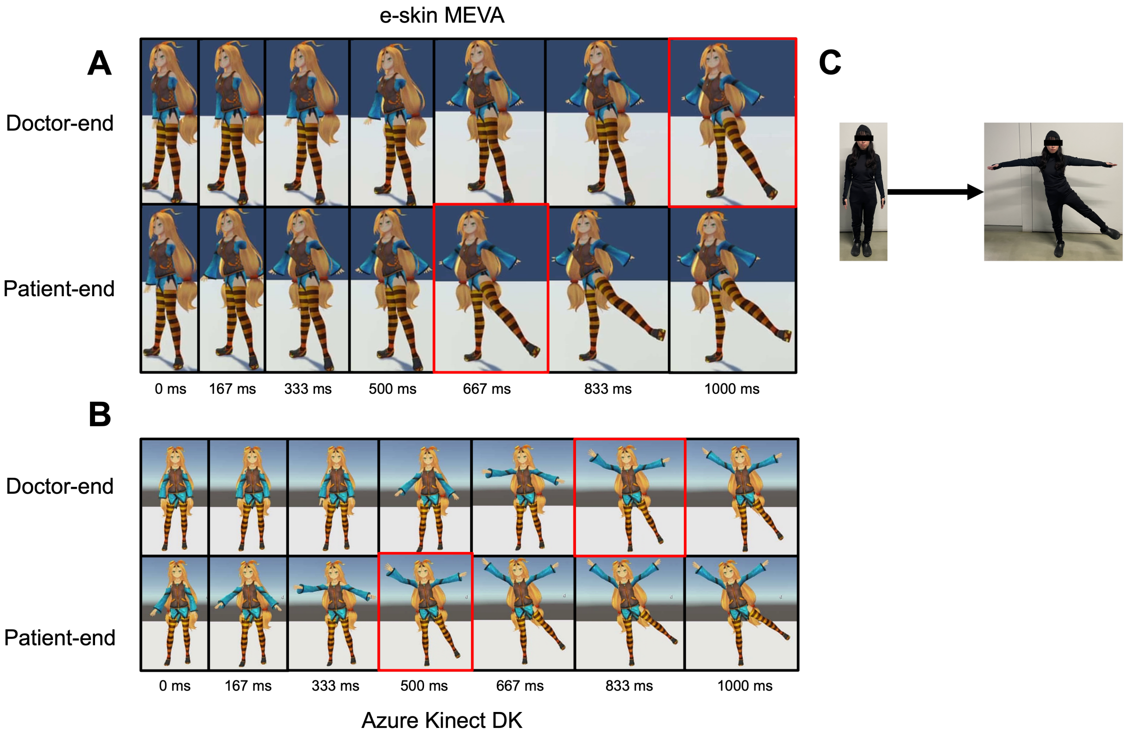

Fig. 3. Demonstration of sample applications.

Sample applications of the RMCP in e-skin MEVA (**A**) and Azure Kinect DK using iPi Studio (**B**). Images of the avatar were captured every 167 ms. The red squares indicate the timing of the left foot moving away from the floor plane. Test users were asked to perform the following pose (**C**). The humanoid avatar is “Unity-Chan” and all rights are reserved by Unity Technologies Japan K. K.