Extraction of spatial information for low-bandwidth telerehabilitation applications

Kok Kiong Tan, PhD;¹ Arun Shankar Narayanan;^{1*} Choon Huat Koh, PhD;² Kevin Caves;³ Helen Hoenig, MD, PhD⁴

¹Department of Electrical and Computer Engineering and ²Saw Swee Hock School of Public Health, National University of Singapore, Singapore; ³Department of Surgery, Duke University Medical Center, Durham, NC; and Biomedical Engineering Department, Duke University, Durham, NC; ⁴Department of Medicine, Duke University School of Medicine, Durham, NC

Abstract—Telemedicine applications, based on two-dimensional (2D) video conferencing technology, have been around for the past 15 to 20 yr. They have been demonstrated to be acceptable for face-to-face consultations and useful for visual examination of wounds and abrasions. However, certain telerehabilitation assessments need the use of spatial information in order to accurately assess the patient's condition and sending three-dimensional video data over low-bandwidth networks is extremely challenging. This article proposes an innovative way of extracting the key spatial information from the patient's movement during telerehabilitation assessment based on 2D video and then presenting the extracted data by using graph plots alongside the video to help physicians in assessments with minimum burden on existing video data transfer. Some common rehabilitation scenarios are chosen for illustrations, and experiments are conducted based on skeletal tracking and color detection algorithms using the Microsoft Kinect sensor. Extracted data are analyzed in detail and their usability discussed.

Key words: Kinect, low-bandwidth video, skeletal tracking, spatial data extraction, telehealth, telemedicine, telemedicine applications, telerehabilitation, telerehabilitation assessment, videoconferencing.

INTRODUCTION

With advances in communication technology, simple but common diagnosis and follow-up medical consulta-

tions for rural populations can be made over the Internet. This technology, termed “telemedicine,” can be most effective in developing countries where access to better medical care is limited. With 84 percent of the global population residing in developing countries, the system designed has to effectively utilize the limited availability of medical resources [1]. The fact that more than 60 percent of total Internet users are from these countries demonstrates that a basic communication infrastructure already exists in these areas, although the maximum network speed may be as low as 128 kb/s in some places [2].

Two-dimensional (2D) video-based medical consultation has been explored widely for the past 15–20 yr. 2D televideo has been demonstrated to be acceptable for face-to-face consultation supplementing telephone and useful for visual examination of wounds, abrasions, etc. [3]. However, certain clinical examinations necessitate

Abbreviations: 2D = two-dimensional; 2D+Z = 2D-plus-Depth; 3D = three-dimensional; D³ = Dynamic Data Display; RGB = red, green, and blue; SDK = software development kit; WPF = Windows Presentation Foundation.

*Address all correspondence to Arun Shankar Narayanan, National University of Singapore–Electrical and Computer Engineering, Blk E4A 03-04, Engineering Drive 3 117576, Singapore; +65 6516 4460; fax: +65 6777 3117.

Email: a0035708@nus.edu.sg

<http://dx.doi.org/10.1682/JRRD.2013.09.0217>

the use of three-dimensional (3D) video for accurate assessment of the patient's condition. One such scenario, as discussed by Hoenig et al. [4], is gait assessment. A 2D video from the frontal view was not sufficient to reliably assess foot placement during walking. Simply having the patient turn sideways was able to compensate for a single-lens camera when measuring cane height. However, this may not be feasible for gait assessment, which typically requires a minimum distance of 3 m for standardized measures of gait and balance [5].

One way to overcome this issue is by transmitting stereoscopic images and processing the images to render a 3D reconstructed video at the receiver by using special algorithms. Welch et al. rendered high-quality 3D video from a remote location by using multiple cameras in an application named 3D medical consultation or "3DMC" [3]. In their prototype, a small array of cameras is used to reconstruct a real-time online 3D computer model of the real environment and events. But the challenge in implementing such a system is the high amount of data incurred while operating over bandwidth-limited networks. Sending 3D videos at network speeds of 128 kb/s is almost impossible. According to experiments conducted by Russell et al. [6], who studied the insufficiency of single 2D camera in gait assessment, two cameras with both frontal and lateral view can fulfill the spatial-resolution requirement. While Russell et al. also showed that this method can work with low-bandwidth internet transmission, it nonetheless required a huge data payload.

Another alternative is to use 2D-plus-Depth (also known as 2D+Z) technology, a stereoscopic video coding format that is used for 3D displays, developed by Philips 3D Solutions (Eindhoven, the Netherlands) [7]. Each 2D frame is supplemented with a depth map captured using a Time-Of-Flight sensor. This depth map is a grayscale map with white indicating the pixel in front of the display and black indicating the pixel in the background. This has the advantage of taking only 5–20 percent more bandwidth than 2D images [8]. However, a disadvantage is the limited amount of depth displayed using 8-bit grayscale. Moreover, creating accurate 2D+Z videos is generally costly and difficult and the depth cannot be reliably estimated in monocular video in most cases. It also needs special software to do the conversion in live mode [9]. There are some systems developed for rehabilitation purposes using Kinect technology (Microsoft Corp; Redmond, Washington) [10], but these are mainly for in-house rehabilitation programs and they do not specifi-

cally address the issue of information transfer over low-bandwidth networks.

In situations where rendering of complete 3D video is impractical, transferring spatial data for only specific features can be considered. This article, written as an add-on to the framework presented previously by the same authors [11], introduces an innovative way of utilizing the depth information of selected objects in the image together with the usual 2D video, thus achieving some degree of spatial resolution by graphical representation of this information in certain telerehabilitation applications. Microsoft Kinect sensor is used together with an open-source color detection and tracking library package to extract the desired depth data from the image. The extracted depth data are then plotted in a graph and displayed alongside the images in order to make sensible conclusions about the patient's condition. Only a few extra bytes are needed to represent the depth data and, hence, it does not induce any significant load over existing 2D video transmission. In this article, examples of telerehabilitation assessment for which spatial resolution is required will be highlighted and solutions leveraging the proposed approach will be presented to show the effectiveness of such approaches.

METHODS

Hardware and Software

The main hardware used in the development of this framework is the Microsoft Kinect sensor (**Figure 1**) with the software development kit (SDK) for Kinect installed in a Windows 7-based computer (Microsoft Corp). An assessment room of at least 3 m in width and length was required to carry out some of the assessments of telerehabilitation.

The programming was done in Visual Studio 2010 (Microsoft Corp) on the Windows Presentation Foundation (WPF) platform using C# language. C# was chosen for its readily established wide development on Kinect-based technology and also the ease in building the user interface for such applications. Kinect skeleton stream was used for tracking specific body parts when required.

Color detection was carried out based on AForge.NET [12], an open-source library developed for C#.Net framework, and uses different kinds of color filters in order to achieve the purpose. All the extracted depth information was presented on a dynamic graph

Figure 1.
Kinect sensor.

panel as a line graph. This graph was embedded in the program using Dynamic Data Display (D^3) [13], an open-source project owned by Microsoft Research. D^3 is a way of visualizing a dynamic data set in the WPF application in the form of a line graph, bar chart, etc. Zooming and panning are already embedded in this library, which helps in obtaining a closer look at the data for deeper analysis. This library also has embedded functions to copy and save the plotted graph to the hard disk by right clicking on the plot.

Depth Extraction Based on Color

As mentioned, AForge.NET open-source library was used for color tracking. There are multiple image processing filters that allow filtering of pixels depending on their color values. These filters may be used to determine whether the specified color is detected in the displayed image frame. For example, *ChannelFiltering* filter filters the red, green, and blue (RGB) values of pixels inside/outside of specific range operating in RGB color space. Another filter named *HSLFiltering* filter operates in hue, saturation and luminance color space and filters those pixels inside a specified range and keeps out the rest of the pixels. In this article, *EuclideanFiltering* filter was used because of its effectiveness and short processing time while operating in the performance test environment. This filter filters those pixels inside of the RGB sphere with a specified center and radius and fills the rest with a specified color.

This filter was applied to Kinect's color image stream for object detection. The color image frame from Kinect's color stream is initially converted to a bitmap image format. This bitmap image is then passed to

another method together with the selected color to run the color detection algorithm based on the filter just described. Once the specified color is detected, the biggest single object with the specific color is identified through an iterative algorithm that loops through all the objects inside the image and a rectangle is drawn around that object. The center of the rectangle is then calculated and a small circle is drawn as well for the user to identify the point of focus. The x- and y-coordinates of this point are used to identify the distance of the object from the Kinect sensor. The new image with the rectangle is converted to a bitmap source object in order to display on the screen. This same image is sent to the receiver after going through appropriate encoding algorithms.

Kinect sensor also has a depth stream that stores the distance data of each pixel on the camera's field-of-view in a 16-bit array format of short integers. A zero value denotes that no data are available at the pixel, either because the object is too close or too far from the camera. The detectable range from the Kinect sensor is from 0.4 to 4.0 m. This depth information is stored in a specific location corresponding to each pixel's coordinates in the depth image frame. For example, if x and y are the coordinate information of the specific pixel in an image frame, the corresponding depth information for that point will be stored at the location given by **Equation 1**:

$$[x + y * width] , \quad (1)$$

where *width* is the width of the depth frame.

Out of the 16 bits used to represent the depth, only the last 13 are used to represent the distance. Therefore, this depth information can be converted to distance in millimeters by doing a right shift operation. Assuming P_i represents the 16-bit depth data at a particular pixel, the distance D_i is calculated with **Equation 2**:

$$D_i = P_i \gg 3 , \quad (2)$$

where $\gg$ denotes the logical right shift operation.

With use of **Equations 1** and **2** and the coordinates of the center of object calculated by matching the depth image frame and the color frame, the distance of the colored object from the Kinect can be extracted. This information is later sent to the line graph using D^3 . This plot will have the detected depth in the y-axis and the data point number on the x-axis. By observing the graph, one can tell how far the object is located from the Kinect sensor.

Depth Extraction Based on Skeleton

The ability to extract and track the positions of the human body is another functionality of the Kinect. Other than the RGB image stream and the depth stream, the Kinect sensor also has a skeleton stream that can detect 20 joints of the skeleton, including the head, hip, knee, and ankle. In some telerehabilitation exercises where a full human skeleton has to be visible on screen, the skeleton stream from Kinect can be utilized to detect certain body positions and assess whether the patient is performing the exercise in the right way.

There are two modes in skeleton tracking: seated mode and default. Default, also known as standing mode, tracks all the 20 skeletal joints; the seated mode only tracks the 10 upper-body joints, including shoulders, elbows, wrists, arms, and head. The default mode detects the user based on the distance of the subject from the background, and the seated mode uses movement to detect the user and distinguish him or her from the background, such as a chair or couch [14]. In cases where the patients have lower-body paralysis, default mode is recommended because the rehabilitation focus will be on the lower-body joints and seated mode will only capture upper-body joints. Thus, different tracking modes are used in this framework based on the consultation requirements.

Depth range can be set to either near mode or default mode. In the near mode, skeletal tracking returns position-only tracked skeletons without the possibility of getting the full skeletal joint positions, from as close to the sensor as 0.4 m up to a maximum of 3.0 m. Although both seated and default modes are usable in the near range mode, the seated mode is more commonly used because in this mode, the whole skeleton does not need to be tracked. In default depth range, the tracking position can be from 0.8 to 4.0 m. Again, the depth range was chosen based on the application requirement in this framework [14].

Similar to the case for color-based detection, the color image frame from the color stream is converted to a bitmap image format. This is for editing of the frame with the identified joint later on. Once an active skeleton is detected using the skeleton stream, the program will identify the specified joints on the detected skeleton. This skeleton point will be marked with a filled red ellipse using a separate method. The new bitmap image with the ellipses drawn will be converted to a bitmap source as in the previous case and sent to the receiver after passing through appropriate encoding algorithms. The depth of the detected joint can be directly extracted using the SDK

commands. This value, in meters, has to be multiplied by 1,000 in order to be converted to millimeters.

Velocity Extraction from Joints

Other motion parameters were derived from the coordinate positions of the joints in the skeleton. By making use of this information, it is possible to measure a motion parameter such as the velocity of a specified joint while the patient performs the exercise. For example, consider the case of walking assessment with the use of skeleton stream by tracking the hip joint position of the patient. The patient walks along a straight line and the physician assesses his or her deviation from the straight line as well as his or her walking speed.

Let the hip position coordinates at time = t_1 be at (x_1, y_1, z_1) and at time = t_2 be at (x_2, y_2, z_2) . Velocity (v) is the time derivative of position, and by using this analogy, the absolute velocity of the hip joint between t_1 and t_2 can be calculated as shown in **Equation 3**:

$$|v| = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}{(t_2 - t_1)} \quad (3)$$

This estimated value of velocity will be in meters per second because the coordinate values are measured in terms of millimeters and the time in terms of milliseconds. The interval between t_1 and t_2 is chosen to be 20 data points. If it is too short, slow movements cannot be measured because the velocity will be close to zero value. If it is too long, fast movements will not be measured well. The sampling rate of 20 data points translates to between 1 and 2 s in the time scale. From repeated trials, this value was found to be most ideal for measuring velocity in most of the exercises carried out by patients. This method was used to measure the velocity of body joints in other exercises as well.

Instead of calculating the absolute velocity, we calculated individual velocity components as well in order to understand the velocity of motion along each direction. The below equations are formulated based on the fundamental principle that velocity is calculated from the rate of change of position. Thus rate of change of position in each direction (x , y , and z) is calculated with **Equations 4–6**:

$$v_x = \frac{(x_2 - x_1)}{(t_2 - t_1)} \quad (4)$$

$$v_y = \frac{(y_2 - y_1)}{(t_2 - t_1)}, \text{ and} \quad (5)$$

$$v_z = \frac{(z_2 - z_1)}{(t_2 - t_1)}. \quad (6)$$

Calculating individual velocity along each axis while carrying out specific exercises (for example, lifting a load while stretching arms) can assist in identifying both the direction of the movement and the direction in which the movement is the weakest and thus improve rehabilitation assessment. Absolute velocity alone would not have been sufficient for such a purpose.

The skeletal joint information can be adjusted across different frames to minimize jittering and stabilize the joint positions, thus stabilizing the calculation of velocity. A smoothing mechanism was provided by Kinect SDK that is based on the Holt Double Exponential Smoothing method [15]. Skeleton stream was enabled with this smoothing filter, which filters out small jitters with minimum latency. This filter can be controlled using five parameters: *Smoothing*, *Correction*, *Prediction*, *JitterRadius*, and *MaxDeviationRadius* [16]. The *Smoothing* parameter must be in the range of 0 to 1, and increasing the value leads to more highly smoothed skeleton positions. As the value increases, the responsiveness decreases. The *Correction* parameter must also be in the range of 0 to 1. Lower values result in slower response to correction toward the raw data and greater smoothing. *Prediction* refers to the number of frames predicted into the future and the value must be greater than or equal to zero. For this parameter, values greater than 0.5 may lead to overshooting when the joints move quickly. *JitterRadius* is the radius in meters for jitter reduction. *MaxDeviationRadius* is the maximum radius in meters that the filtered positions are allowed to deviate from the raw data. These values were chosen such that there was little latency incurred by the filter while small jitters were effectively filtered out [16].

Calculation of Angle Between Joints

The coordinate information for each joint can be used for calculating the angle between them. For example, if the coordinate data for right wrist, right elbow, and right shoulder are known, the angle between upper and lower arm with elbow as center point can be calculated using

vector manipulation. To achieve this, the first vector is formed from right elbow to right wrist, let it be A , and the second vector from right elbow to right shoulder, let it be B , using vector calculation using the corresponding coordinate data for each joint. The angle between the two vectors can be measured using **Equation 7**:

$$\cos\Theta = \frac{A \cdot B}{|A| * |B|}, \quad (7)$$

where the numerator is the dot product of the two vectors and the denominator is the product of their magnitudes. Such calculations can be used to assess the patient's exercise pattern (e.g., lifting a load with the right hand) so that the physician can observe whether the change in angle is uniform throughout the lifting process and detect the position at which the hand is weaker or stronger during the exercise.

RESULTS

A few common assessment scenarios will be used in this section to leverage the tracking and representation of depth data of specific body parts or objects. Test runs were conducted on these scenarios using the methods explained earlier, and the data obtained are represented in line graphs. Tests were categorized as cases that tracked one mobile object, one mobile and one fixed object, two mobile objects, multiple joints of human skeleton, and the velocity and angle of joint movements. This section provides the details of the test runs and analysis of the collected data.

Tracking Single Mobile Object

In this test, a single mobile object is tracked based on user-specified color. One possible area of application is the assessment of the walking posture of a patient. In a normal 2D video, it is not possible for the physician to see whether the patient is walking on a straight line and at the same time assess the body posture of the subject by looking from a lateral view or a frontal view alone. This spatial resolution framework can be useful in such an application by tracking the depth information of the subject's body while he or she carries out the exercise.

The Kinect, placed on a table top as parallel to the ground as possible, will give feedback on the patient's position based on the color tracking of the patient's

jacket. The subject was asked to walk along a straight line of 2.15 m as parallel to the Kinect as possible. **Figure 2** shows this scenario. The user can select the specific color to be tracked on the screen. In this case, the subject's jacket color is selected. The program draws a rectangle around the captured object and a circle to mark the point of measurement. The returned depth data from the Kinect are plotted on a line graph on the right. The x-axis represents the data point number whereas the y-axis is the measured depth in millimeter.

A moving average is calculated based on the previous 100 measured points and plotted along the measured values in order to assist in the assessment. A “first-in, first-out” queue is used to achieve this function. Once the first 100 points are available, a new loop starts to operate that will calculate the average of the available data. The newly arrived data will push the most dated data out of the queue and the average is calculated again. This process is continuous. Programming logic for the above-mentioned algorithm is drawn in **Figure 3**. This average plot will help the physician determine how much the patient has deviated from the straight line while carrying out the designated exercise, as well as understand the underlying trend of the plotted data.

Based on the projected data line, the physician can determine whether the exercise was performed correctly. For example, the depth data within a range of values

from the average line implies that the walk was along a straight line, whereas a wide data fluctuation implies that the walk was in a zigzag line. Two separate trials were recorded and plotted in **Figure 4**. “Good” represents a patient walking on the line and “Bad” represents a patient walking in a zigzag line. The color tracking method is used instead of the skeleton-based approach in this example because the full human skeleton may not be necessarily visible at all times in this scenario.

Measurement error of Kinect's depth sensor was determined to be 2 mm at 1 m depth and 2.5 cm at 3 m depth, according to a study by Khoshelham and Elberink [17]. In this exercise, the patient was 2 m away from Kinect, which meant that the measurement accuracy was within 1–2 cm and the Kinect was still able to detect movements outside this range accurately. This suggests that the assessment performed in this case was reliable.

Tracking Single Mobile Object with Reference to Fixed Object

In this case, two objects were being tracked: a fixed object and a mobile object. To select the fixed object, the user has to click a mouse over the object and a red circle will be drawn around the selected point. The depth of this object is then calculated based on the coordinate values of this position, as explained in the previous section. The

Figure 2.
Single object tracking program screen.

Figure 3. Programming algorithm for moving average calculation.

mobile object is chosen based on color as in the previous case. In this case, the subject is asked to perform a simulation of drinking water from a cup. The subject's head is chosen as the fixed position, and the yellow cup is selected as the mobile object.

Figure 5 shows the experiment setup in the program window. Both the returned depth data are plotted on the graph on the right. From the line graph plot, the physician can see how consistently the exercise was performed by the subject by comparing both plotted lines. **Figure 6** shows a magnified image of the graph plot for further analysis by the physician if necessary. If the cup is closer to the mouth, both lines will be close and vice versa. Analyzing the graph carefully can reveal data such as how many cycles were done by the patient in a particular time window and can even be used as a gauge to measure the patient's progress over multiple sessions of telerehabilitation.

Tracking of Single Mobile Object with Reference to Moving Body Part

One drawback to the method of selecting a body part as a fixed object is that once the body part moves, the reference depth is changed, although it may not affect the final assessment of the exercise. Another drawback is that if any other object is brought to the front and blocks the selected point, the new object's depth will be captured. To overcome this limitation, two methods are proposed. One is to track a body part based on skeleton stream from the Kinect. The other is to track based on color of the fixed object. The former method is explored in detail here and the latter method will be explained in the next case.

In the previous exercise, the subject's body part tracked was the subject's body position. Thus, Kinect's skeletal stream was activated to track the head in this case. Since the full body skeleton was not visible to Kinect in this case, the "seated" mode was activated with the depth range set to "near." Once the upper body was visible to Kinect, the selected body part was tracked and marked by a filled red ellipse. The head was selected here. The other mobile object tracked was selected by specifying the color as in previous cases. **Figure 7** shows the program window of this exercise and **Figure 8** shows the data being tracked for further analysis.

Tracking Multiple Mobile Objects Based on Color

In the previous cases, there was only one mobile object and the other was in fixed position. In this case, the program was designed to track two mobile objects by choosing the object color to be tracked. The program identified the objects with the specified colors and drew a rectangle around each of them, and Kinect fed back the depth information of those detected objects.

In **Figure 9**, the subject was doing an arm-stretching exercise wearing two gloves of different colors (pink and yellow). The subject was asked to move both arms toward and away from the body while facing Kinect. Kinect captured the objects' depth data and plotted them in the graph on the right. Analyzing the enlarged graph in **Figure 10** helped the physician to see whether both arms were being stretched equally and whether they were synchronized during the exercise. As mentioned in the previous case, one can see how many cycles of stretching is performed in a fixed time window and whether the patient was improving as the telerehabilitation session progressed.

Figure 4.

Results of walking posture analysis: **(a)** first trial and **(b)** second trial. “Good” = patient walked on line and “Bad” = patient walked in zigzag line.

Detecting and Tracking of Multiple Joints

In some exercises, human skeleton-based assessment can be very useful to the physician, such as during cane height assessment, balancing tests such as double-leg stance and single-leg stance [18], and even finger-to-nose task performance tests [19]. The user can select the joints to be tracked and plot the corresponding depth information on a graph.

In the first case here, the subject was asked to walk with the cane while the physician assessed the walking posture and determined whether the cane height was ideal for the patient. Since the whole body had to be visible to Kinect, “Default” mode was used instead of “Seated” for the skeletal stream. The depth range was set to “Default” as well, which enabled Kinect to track the body joints to the maximum distance away from the sensor. **Figure 11** shows the positioning of the Kinect with

reference to the subject and the joints being tracked. Kinect was facing the subject holding the cane, and head and hip were chosen to be tracked in this particular case.

Figure 12 shows the program window with the subject holding the cane and the selected joint positions detected by Kinect and identified by the red ellipses. The depth data were drawn as a line graph on the right. While analyzing the enlarged graph data in **Figure 13**, one can clearly see whether the subject stands straight or is leaning forward while holding the cane. Close positioning of the lines implies that both head and hip are on the same plane and the subject is standing straight. As the subject leans forward and walks, it can be seen that the hip position moved away from the head plane and thus the subject’s posture is not upright. A similar method can be used for balancing tests to measure the subject’s body movements while balancing on one leg.

Figure 5.
Single mobile object with fixed object tracking program screen.

Figure 6.
Magnified results of drinking simulation exercise.

With a normal 2D video, it is not easy for the physician to make such assessments unless the patient is asked to walk from a lateral point of view with respect to the camera. The problem in this situation is that the alignment of the patient's other body parts, such as shoulder positioning, is not clearly seen. Thus, depth tracking of certain joints helps in this scenario. Another assessment that can be efficiently done using this method is the finger-to-nose test. In this test, the subject's head and right or left hand were chosen to be tracked while he or she carried out the

exercise as shown in **Figure 14**. Analyzing the plotted graph helps to identify the number of exercise cycles carried out by the patient as well as whether the exercise was carried out properly (e.g., whether the hand touched the nose).

Skeleton tracking can also be used for the exercise discussed in the section "Tracking Single Mobile Object" (i.e., assessing the case of walking in a straight line) only if the whole body is visible to Kinect. It can be modified to track only the hip position or any other joint position as desired by the physician. As explained in the section "Velocity Extraction from Joints," the absolute velocity of Joint1 during the walk was calculated and displayed on screen (**Figure 12**).

Velocity and Angle Tracking of Joint During Exercise

In this experiment, the velocity measurement algorithm presented in the section "Velocity Extraction from Joints" was used to measure the velocity of Joint1 and the angle between the shoulder, elbow, and wrist joints was calculated as explained in the section "Calculation of Angle Between Joints." The exercise was performed over four segments as shown in **Figure 15**. The subject started from position 1 and stretched the arm to position 2 (180° or 3.14 radians) in segment 1. In segment 2, the arm goes back to 90° (1.55 radians) as in position 3. Segment 3 is folding the arm fully so that angle goes closer to zero as

Figure 7.
Tracking head position together with mobile object.

Figure 8.
Enlarged image of plotted data for analysis.

shown in position 4. Finally, in segment 4, the arm goes back to position 3.

This movement cycle is repeated and the velocity of the moving hand as well as the angle formed by the hand were captured in real time and plotted on the graph as shown in **Figure 16**. The four segments are marked in the graph and the unit used for angle was radians and that for absolute velocity was meters per second. The plot shown is for the case when the motion is uniform throughout the exercise, which may be an indication of a normal move-

ment. The velocity can be seen changing cyclically from zero to maximum value during smooth arm movement.

Figure 17 shows the graph plot of an exercise cycle carried out by another patient. While analyzing the plotted data, one can see that the stretching part of the motion (segments 1 and 2) was carried out slower than the folding part of the motion (segments 3 and 4). Correspondingly, the absolute velocity plot also showed the same pattern. Velocity spikes are seen in the folding segments, which indicate a quicker motion, whereas lower values were measured at other times. This indicates that the patient has some trouble stretching the arm with ease and this pattern may not be clearly visible with a usual 2D video. Thus, this method may be useful for the physician's assessment.

Delay Incurred by Depth Tracking

Although these measurements are useful and enhance telerehabilitation assessments, they incur processing delay while extracting the depth data. More specifically, the majority of the delay occurs from processing the color tracking algorithm. Although the color filter being used in this framework is the fastest of all currently available, it is not immune to delay. The **Table** shows the average measured delay in all the five cases discussed in the "Results" section. It can be seen that using skeletal tracking induces almost no delay, while processing of the color tracking algorithm accounts for the majority of the delay.

Figure 9.
Tracking multiple objects based on color.

Figure 10.
Enlarged image of data plot.

Kinect's skeleton stream works in parallel with the color image stream, and they both operate at the same frame rate. Both streams are fired at the same instance as well, and this is the reason for having almost no delay when tracking is based on skeleton joints. In the case of tracking based on color, the program takes the color frame first and then performs the color detection algorithm. These two processes cannot be done in parallel, and this is the cause for the delay in all other cases in the **Table**. However, this delay is still negligible in a regular videoconferencing session. In the color tracking scenario,

Figure 11.
Kinect's positioning for cane height assessment.

each color frame has to undergo the tracking algorithm individually and the tracked object's position is attached together with the frame in the form of a rectangle or an ellipse before sending them to the receiver as discussed in the "Results" section. Because each frame is sent to the receiver side individually using a common "timer" function, it is automatically synchronized with the depth

Figure 12.
Tracking of body joints in cane height assessment.

Figure 13.
Analysis of tracked data.

data and thus, there are no issues with depth data arriving out of sync with the video data.

DISCUSSION AND CONCLUSIONS

A framework for extracting the depth information in a telerehabilitation video and transmitting it together with 2D images is proposed and designed in this article. Different methods of tracking specific objects based on color

and human skeleton using Microsoft Kinect were discussed in detail. Multiple scenarios of telerehabilitation exercises leveraging the framework were explored. Collected depth data were analyzed for the assessment of exercises carried out by the patient, and further possibilities of utilizing this data were discussed. This framework can be a very useful tool for getting a good sense of spatial resolution in situations when conventional 3D video transmission can be extremely challenging because of limited network bandwidth.

There are certain limitations in the current framework, and further work is needed to improve the effectiveness of the overall system. Currently, the data analysis has to be done manually by the physician and this can be tedious if done over a long period of time. Improvements can be made to this platform by having the software automatically analyze extracted data and provide quantitative measures on the effectiveness of the exercise being carried out (e.g., by measuring the stretch distance as well as the number of arm stretches in the exercise discussed in the section “Tracking Multiple Mobile Objects Based on Color” or measuring number of cycles carried out in the finger-to-nose task performance assessment). Developing an algorithm to analyze the extracted joint data, similar to the existing methods for extraction of electrocardiogram parameters from the raw sensor data, to automate the performance analysis of the patient will be the main focus for future work on this project.

Figure 14.
Finger-to-nose assessment using multiple joint tracking method.

ACKNOWLEDGMENTS

Author Contributions:

Study concept and design: K. K. Tan, A. S. Narayanan, H. Hoenig, C. H. Koh, K. Caves.

Drafting of manuscript: K. K. Tan, A. S. Narayanan.

Critical revision of manuscript: K. K. Tan, A. S. Narayanan, H. Hoenig, C. H. Koh, K. Caves.

Study supervision: K. K. Tan, H. Hoenig.

Financial Disclosures: The authors have declared that no competing interests exist.

Funding/Support: This material was unfunded at the time of manuscript preparation.

REFERENCES

1. The World Bank. World population growth [Internet]. Washington (DC): The World Bank; [cited 2013 Jul 3]. Available from: http://www.worldbank.org/depweb/beyond/beyondco/beg_03.pdf
2. International Telecommunication Union. ICT facts and figures [Internet]. Geneva (Switzerland): ITU; [2011; cited 2013 Jul 3]. Available from: <http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2011.pdf>
3. Welch G, Sonnenwald DH, Mayer-Patel K, Yang R, State A, Towles H, Cairns B, Fuchs H. Remote 3D medical consultation. Proceedings of the 2nd International Conference on Broadband Networks (BroadNets 2005); 2005 Oct 3–7; Boston, MA. New York (NY): IEEE; 2005. p.1026–33.
4. Hoenig H, Tate L, Dumbleton S, Montgomery C, Morgan M, Landerman LR, Caves K. A quality assurance study on the accuracy of measuring physical function under current conditions for use of clinical video telehealth. Arch Phys Med Rehabil. 2013;94(5):998–1002. [PMID:23337425] <http://dx.doi.org/10.1016/j.apmr.2013.01.009>
5. Faber MJ, Bosscher RJ, van Wieringen PC. Clinimetric properties of the performance-oriented mobility assessment. Phys Ther. 2006;86(7):944–54. [PMID:16813475]
6. Russell TG, Jull GA, Wootton R. The diagnostic reliability of Internet-based observational kinematic gait analysis. J Telemed Telecare. 2003;9(Suppl 2):S48–51. [PMID:14728760] <http://dx.doi.org/10.1258/135763303322596255>
7. Philips 3D Solutions. 3D interface specifications white paper [Internet]. Amsterdam (the Netherlands): Philips;

Figure 15.
Exercise cycle.

Figure 16.
Velocity and angle tracking with four segments highlighted.

Figure 17.
Simulated image of plotted data for analysis.

Table.
Delay encountered in all scenarios in “Tracking Single Mobile Object” section in main text.

Scenario	Average Delay (ms)
Tracking Single Mobile Object	60
Tracking Single Mobile Object with Respect to Fixed Object	70
Tracking Single Mobile Object with Reference to Moving Body Part	75
Tracking Multiple Mobile Objects Based on Color	130
Detecting and Tracking Multiple Joints	5

[cited 2013 Jul 3]. Available from: <http://www.business-sites.philips.com/shared/assets/global/Downloadablefile/Philips-3D-Interface-White-Paper-13725.pdf>

8. Choi J, Min D, Sohn K. 2D-plus-depth based resolution and frame-rate up-conversion technique for depth video. *IEEE Trans Consumer Electronics*. 2010;56(4):2489,2497.

9. Coll B, Faisal Ishtiaq, Kevin O’Connell. 3D TV at home: Status, challenges and solutions for delivering a high quality experience. *Proceedings of the International Workshop on Video Processing and Quality Metrics for Consumer Electronics (VPQM)*; 2010 Jan 13–15; Scottsdale, AZ.

10. Virtual reality Kinect rehabilitation. Kinect rehabilitation with biofeedback [Internet]. Gliwice (Poland): BRONTES PROCESSING Sp. z o.o. (Ltd.); [cited 2013Jul 3]. Available from: <http://www.virtual-reality-rehabilitation.com/>

11. Arun SN, Lam WC, Tan KK. Innovative solution for a tele-medicine application. *Proceedings of the 2012 7th International Conference for Internet Technology and Secured Transactions*; 2012 Dec 10–12; London, United Kingdom.

12. AForge.NET [Internet]. [cited 2013 Jul 3]. Available from: <http://www.aforgenet.com/>

13. Microsoft Corp. Dynamic data display [Internet]. Redmond (WA): Microsoft Corp; [updated 2011 Dec 13; cited 2013 Jul 3]. Available from: <https://dynamicdatadisplay.codeplex.com/>

14. Microsoft Developer Network. Skeletal tracking [Internet]; Redmond (WA): Microsoft Corp; [cited 2013 Jul 3]. Available from: <http://msdn.microsoft.com/en-us/library/hh973074.aspx>

15. Gardner ES Jr. Exponent smoothing: The state of the art. *J Forecast*. 1985;4:1–28.
<http://dx.doi.org/10.1002/for.3980040103>
16. Microsoft Network Developer. Joint filtering [Internet]. Redmond (WA): Microsoft Corp; [cited 2013 Jul 2]. Available from:
<http://msdn.microsoft.com/en-us/library/jj131024.aspx>
17. Khoshelham K, Elberink SO. Accuracy and resolution of Kinect depth data for indoor mapping applications. *Sensors (Basel)*. 2012;12(2):1437–54. [PMID:22438718]
<http://dx.doi.org/10.3390/s120201437>
18. Boonstra TA, Schouten AC, van der Kooij H. Identification of the contribution of the ankle and hip joints to multi-segmental balance control. *J Neuroeng Rehabil*. 2013;10:23.
19. Lanzino DJ, Conner MN, Goodman KA, Kremer KH, Petkus MT, Hollman JH. Values for timed limb coordination tests in a sample of healthy older adults. *Age Ageing*. 2012;41(6):803–7. [PMID:22743152]
<http://dx.doi.org/10.1093/ageing/afs070>

Submitted for publication October 1, 2013. Accepted in revised form January 10, 2014.

This article and any supplementary material should be cited as follows:

Tan KK, Narayanan AS, Koh CH, Caves K, Hoenig H. Extraction of spatial information for low-bandwidth telerehabilitation applications. *J Rehabil Res Dev*. 2014;51(5):825–40.

<http://dx.doi.org/10.1682/JRRD.2013.09.0217>

ResearcherID/ORCID: Kok Kiong Tan, PhD: G-6621-2013; Arun Shankar Narayanan: K-2878-2013