Demonstrating Kinect2-based Spinal Pose Visualization During At-home Physical Therapy Exercise

I. Methodology & Setup

We built a system to support real-time visualization of a 3D spine model that reflects the patient's spinal pose during physical therapy (PT) exercises, using as the input the data captured by a low-cost Kinect V2 depth camera. While the accuracy remains to be further improved, initial results suggest that this could provide a viable solution. See the demo video in the provided link.

A. Device and Software Setting

In the proposed system, a Kinect V2 depth-camera working with Kinect for Windows SDK 2.0, is placed on a table of about 0.6m tall facing towards the patient. During the execution of PT exercises, the system captures the body motion by estimating the 3D cartesian coordinates (x, y, z) of all 25 joints with respect to the camera as output. In the camera 3D space, the coordinates are measured in meters, where $(0, 0, 0)$ is at the center of the sensor, $x > 0$ when the joint is on camera's left side, and $y > 0$ when it is above the camera. z is always positive as it extends out from the camera. Joints placed more than 5m horizontally and 4.5m vertically away from the camera may not be properly detected. In this work, a patient is required to stand at least about 1.5m in front of the camera due to its limitation of the field of view, and no further than 4.5m away to be in the range of effective skeleton tracking.

We use "Spine Viewer", which is a 3D Unity model developed by iSO-FORM[1]. It animates in three axes within the range of 0-60 degree for flexion/extension, lateral bending and axial rotation in both directions and it takes a sequence of 3-tuple (v_1, v_2, v_3) as input to be animated: v_1 describes left lateral bending or right lateral bending; v_2 describes forward flexion or backward flexion/extension of the spine; v_3 describes clockwise axial rotation or anti-clockwise axial rotation.

To achieve the visualization on a Windows PC, we built our application with KinectV2-Unity[2], which is a Unity library enabling Kinect SDK 2.0 package to run on the Unity platform with our 3D model. For visualization, we developed a Python application showing the camera view of the true PT execution from the patient with the captured skeleton joints highlighted.

B. Spine Approximation and PT Exercise

Among the 25 skeleton joints on the body detected by Kinect V2, we pick $Neck$ joint which is the nearest joint to C7 vertebra on a true spine and $Spine\_Base$ joint which is the nearest joint to S2 or S3 vertebra. These vertebras are recognized to be essential to describe the motion of the spine [1]–[3]. We approximate a true human spine as a spine line connecting $Neck$ joint and $Spine\_Base$ joint.

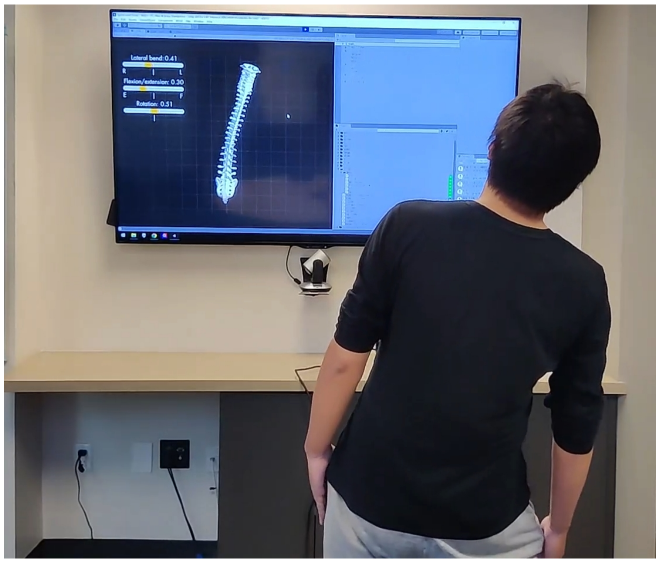

Fig. 1. Motion data captured by a Kinect V2 depth camera from the PT execution of a patient are processed and then provided to the 3D Unity spine model for real-time animation.

This spine approximation presents spine flexion and bending but not spine twisting. To this end, we pick $Shoulder\_Left$ joint and $Shoulder\_Right$ joint and connect them to approximate the shoulder of a patient. We assume patients follow the instructions of PT exercise in [4] and keep the shoulder line perpendicular to the neck and to the direction of trunk as well. Therefore, the rotation of shoulder line projection on plane XOZ can be viewed as spine twisting angle. More details on 3D angle computation are discussed in Section II-A.

Given the fact that there is no evidence to show one type of exercise is superior to another [5], the proposed work focuses more on PT exercises that are well defined and easy for patients to learn and execute at home. Therefore, we strictly follow the exercises introduced and defined in [3, 4] as following:

- Neutral Posture
- Flexion/Extension
- Left/Right Lateral Bending
- Clockwise/Anti-Clockwise Axial Rotation

[1]https://www.iso-form.com/
[2]https://github.com/clarte53/kinectv2-unity

II. Algorithm

This section presents the key algorithmic steps behind the demo system, focusing on how the 3D spine model is estimated based on the 3D coordinates in the camera space.

A. 3D Angle Computation

We define the coordinates of $Neck$, $Spine\_Base$, $Shoulder\_Left$ and $Shoulder\_Right$ joints in current state as $(x_{neck}, y_{neck}, z_{neck})$, $(x_{base}, y_{base}, z_{base})$, $(x_{left}, y_{left}, z_{left})$ and $(x_{right}, y_{right}, z_{right})$ respectively, hence the vector of spine approximation and shoulder approximation are computed as:

$$\overrightarrow{V}_{spine} = (x_{neck} - x_{base}, y_{neck} - y_{base}, z_{neck} - z_{base}) \quad (1)$$

$$\overrightarrow{V}_{shoulder} = (x_{right} - x_{left}, y_{right} - y_{left}, z_{right} - z_{left}) \quad (2)$$

Then the projection of the spine vector on plane YOZ and XOY, and the projection of the shoulder vector on plane XOZ can be easily obtained as:

$$\overrightarrow{VP}_{spineYOZ} = (0, y_{neck} - y_{base}, z_{neck} - z_{base}) \quad (3)$$

$$\overrightarrow{VP}_{spineXOY} = (x_{neck} - x_{base}, y_{neck} - y_{base}, 0) \quad (4)$$

$$\overrightarrow{VP}_{shoulderXOZ} = (x_{right} - x_{left}, 0, z_{right} - z_{left}) \quad (5)$$

If we define $\overrightarrow{E_x}$ as a unit vector along X axis and $\overrightarrow{E_y}$ as a unit vector along Y axis, then the flexion/extension angle on plane YOZ is the angle between $\overrightarrow{VP}_{spineYOZ}$ and $\overrightarrow{E_y}$; the lateral bending angle on plane XOY is the angle between $\overrightarrow{VP}_{spineXOY}$ and $\overrightarrow{E_y}$; the axial rotation angle on plane XOZ is the angle between $\overrightarrow{VP}_{shoulderXOZ}$ and $\overrightarrow{E_x}$. Generally, if we define the angle between a vector $\overrightarrow{VP}$ and a vector $\overrightarrow{E}$ as θ, we have:

$$\cos\theta = \frac{\overrightarrow{VP} \cdot \overrightarrow{E}}{\left|\overrightarrow{VP}\right| \times \left|\overrightarrow{E}\right|} \quad (6)$$

where $|VP|$ and $\left|\overrightarrow{E}\right|$ are the magnitude of those vectors. Therefore, the flexion/extension θ_{flex}, lateral bending θ_{bend} and axial rotation θ_{rot} angles are computed as:

$$\theta_{flex} = \arccos \frac{\overrightarrow{VP}_{spineYOZ} \cdot \overrightarrow{E_y}}{\left|\overrightarrow{VP}_{spineYOZ}\right| \times \left|\overrightarrow{E_y}\right|} \quad (7)$$

$$\theta_{bend} = \arccos \frac{\overrightarrow{VP}_{spineXOY} \cdot \overrightarrow{E_y}}{\left|\overrightarrow{VP}_{spineXOY}\right| \times \left|\overrightarrow{E_y}\right|} \quad (8)$$

$$\theta_{rot} = \arccos \frac{\overrightarrow{VP}_{shoulderXOZ} \cdot \overrightarrow{E_y}}{\left|\overrightarrow{VP}_{shoulderXOZ}\right| \times \left|\overrightarrow{E_y}\right|} \quad (9)$$

We want to identify not only the values of 3D angles but also the direction of spine motions. Specifically, given the computed $\theta_{flex} = 30°$, we need to determine if it is a $30°$ forward flexion or backward extension. To this end, we slide the $\overrightarrow{V}_{spine}$ and $\overrightarrow{V}_{shoulder}$ to the origin location in the camera 3D space without changing its direction or length and define the motion in following rules:

- If z value of $\overrightarrow{V}_{spine}$ is smaller than 0, we define $\theta_{flex} < 0$ and the current θ_{flex} is a forward flexion angle.
- If x value of $\overrightarrow{V}_{spine}$ is smaller than 0, we define $\theta_{bend} < 0$ and the current θ_{bend} is a left lateral bending angle.
- If z value of $\overrightarrow{V}_{shoulder}$ is smaller than 0, we define $\theta_{rot} < 0$ and the current θ_{rot} is an anti-clockwise axial rotation angle.
- Left-hand side of a patient is on right side of the camera space.

B. 3D Spine Model Animation

As mentioned in Section 2, Spine Viewer (by iSO-FORM) requires a sequence of (v_1, v_2, v_3) to be animated. Each of the value in this 3-tuple represents a position on a slide as length of 1 to describe the motion of lateral bending, flexion/extension and axial rotation. The range of each slider value is from 0 to 1 and 0.5 is the neutral posture. The spine model animates based on the following rules:

- If $0 \leq v_1 < 0.5$, the spine model bends towards right-hand side.
- If $0 \leq v_2 < 0.5$, the spine model extends backward.
- If $0 \leq v_3 < 0.5$, the spine model twists clock-wise.

Since the maximum bending and twisting angle of the model is $60°$, we compute (v_1, v_2, v_3) as:

$$v_1 = 0.5 - 0.5 \times (\theta_{bend}/60) \quad (10)$$

$$v_2 = 0.5 - 0.5 \times (\theta_{flex}/60) \quad (11)$$

$$v_3 = 0.5 - 0.5 \times (\theta_{rot}/60) \quad (12)$$

References

[1] J. E. Caviedes, B. Li, and V. C. Jammula, "Wearable sensor array design for spine posture monitoring during exercise incorporating biofeedback," *IEEE Transactions on Biomedical Engineering*, vol. 67, no. 10, pp. 2828–2838, 2020.

[2] M. Betsch, M. Wild, P. Jungbluth, M. Hakimi, J. Windolf, B. Haex, T. Horstmann, and W. Rapp, "Reliability and validity of 4d rasterstereography under dynamic conditions," *Computers in biology and medicine*, vol. 41, no. 6, pp. 308–312, 2011.

[3] J. Caviedes, B. Li, P. Swan, and J. Chen, "A new wearable stretch sensor array for 3d spine model visualization during therapeutic exercise," *Medical Engineering & Physics*, vol. 99, p. 103732, 2022.

[4] J. Chen, J. Caviedes, and B. Li, "Classification of single-axis spinal motion using a wearable system of stretch sensors for at-home physical therapy," in *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2021, pp. 7404–7407.

[5] N. E. Foster, J. R. Anema, D. Cherkin, R. Chou, S. P. Cohen, D. P. Gross, P. H. Ferreira, J. M. Fritz, B. W. Koes, W. Peul *et al.*, "Prevention and treatment of low back pain: evidence, challenges, and promising directions," *The Lancet*, vol. 391, no. 10137, pp. 2368–2383, 2018.