CEA Comité Español de Automática

XLV JORNADAS AUTOMÁTICA MÁLAGA 2024

UNIVERSIDADE DA CORUÑA

Jornadas de Automática

Geodesic restricted aruco-based positioning for VR rehabilitation robotics

Montesino, I.*, Alonso, H., Victores, J.G., Balaguer, C., Jardón, A.

[a]*Department of Systems Engineering and Automation, Universidad Carlos III de Madrid, Avda. de la Universidad 30, Leganés, 28911, Madrid, Spain*

To cite this article: Montesino, I., Alonso, H., Victores, J.G., Balaguer, C., Jardón, A. . 2024. Geodesic restriced aruco-based positioning for VR rehabilitation robotic. Jornadas de Automática, 45.
https://doi.org/10.17979/ja-cea.2024.45.10964

Resumen

La demanda de rehabilitación y terapia física en el mundo es cada vez mayor, debido a un envejecimiento de la población y a una vida más sedentaria. Una de las terapias más prometedoras es la terapia asistida con realidad virtual (VR), ya que mejora el ánimo del paciente, su autonomía y con ello las horas que le dedica a la terapia. Las carencias de este tipo de terapias surgen de no poder ofrecer ejercicios resistivos o con ayuda física al paciente. En el proyecto Roboasset hemos desarrollado un robot para simular ejercicios físicos en un entorno VR. Una problemática todavía no resuelta es conseguir la paridad de los elementos virtuales con los físicos. En este artículo desarrollamos un método para sincronizar los sistemas de coordenadas del robot físico con los del mundo virtual mediante el uso de marcadores fiduciarios (ArUco) y una cámara externa. Usando las propiedades geométricas de los cuaternios, conseguimos que la sincronización de los sistemas de coordenadas no introduzca ningún ruido en el ángulo del suelo.

Palabras clave: Trabajo en entornos reales y virtuales, Ingeniería de rehabilitación y prestación de servicios de salud, Percepción y sensado, Tecnología asistencial e ingeniería de rehabilitación, Manipuladores robóticos, Fusión de información y sensores, Computación centrada en el ser humano, Interfaces inteligentes,

Geodesic restricted aruco-based positioning for VR rehabilitation robotics

Abstract

The demand for rehabilitation and physical therapy worldwide is increasing, due to an aging population and a more sedentary lifestyle. One of the most promising therapies is virtual reality-assisted therapy (VR), as it improves the patient's mood, autonomy, and consequently, the time devoted to therapy. Shortcomings of this type of therapy arise from the inability to offer resistive exercises or physical assistance to the patient. In the Roboasset project, we have developed a robot to simulate physical exercises in a VR environment. An unresolved issue is achieving parity between virtual and physical elements. In this article, we develop a method to synchronize the coordinate systems of the physical robot with those of the virtual world using fiducial markers (ArUco) and an external camera. By leveraging the geometric properties of quaternions, we ensure that the synchronization of coordinate systems does not introduce any noise in the ground angle.

Keywords: Work in real and virtual environment, Rehabilitation engineering and healthcare delivery, Perception and sensing, Assistive technology and rehabilitation engineering, Robots manipulators, Information and sensor fusion., Human-centered computing, Intelligent interfaces.

1. Introduction

Conventional rehabilitation therapies often face challenges such as low adherence and lack of motivation in patients, which can affect their recovery process (Geest et al., 2003). In this context, gamification-based therapies have proven effective in improving patient continuity and engagement with treatment, accelerating their recovery (Steiner et al., 2020; Veerbeek et al., 2017).

*Autor para correspondencia: imontesi@ing.uc3m.es

To enhance the benefits of gamified therapies, the inclusion of virtual reality (VR) has represented a significant advancement. VR offers unique possibilities for creating and designing interactive environments that promote patient immersion. Devices like Meta Quest allow for the development of realistic experiences at an accessible cost, increasing patient interaction and creating an extension of their reality. These immersive experiences allow patients to temporarily regain lost skills or abilities, improving their rehabilitation process (Lewis and Rosie, 2012; Laver et al., 2015). However, one of the challenges of VR is the lack of physical interaction, a fundamental characteristic for performing exercises that involve dexterity or strength. To address this issue, elements that provide haptic feedback during interaction with virtual objects, such as sensors and motors, can be incorporated (Bardorfer et al., 2001). Nevertheless, the most effective solution would be to integrate real elements into the virtual space, allowing their manipulation in both realities. To achieve this, it is necessary to synchronize the real and virtual worlds.

The synchronization of real and virtual elements requires precise calibration that compares the positions of real objects within the virtual environment. For a long time studies have shown the high level of sensitivity that humans have for horizontal tilt (Vogels et al., 1985) (Ceyte et al., 2009) and their effect on visualmotor learning (Jiang et al., 2018). With humans being sensitive to deviations as low as a few degrees, even slight changes in visual cues can significantly influence our perception of gravity and self-orientation (Barnett-Cowan et al., 2008).

Computer vision is a crucial tool in this process, allowing the recognition of points of interest in the environment, both natural and artificial. Fiducial markers like ArUco and AprilTags facilitate accurate tracking of the position and orientation of objects (Kalaitzakis et al., 2021). To implement this detection, additional hardware is needed, as access to the images captured by VR glasses is not available. One possible solution is the placement of a depth-sensing camera fixed to the device (Michiels et al., 2024; Reimer et al., 2021). In our case, we propose using an external camera intended for postural detection.

Our solution involves the use of two markers: one attached to the front of the VR headset and another fixed at a static point in the environment. Using the detection provided by OpenCV and calculating transformation matrices between various real and virtual elements, we seek to position the user's point of view (POV) within the immersive environment. This will allow the user to interact with virtual elements that coincide with real elements, enhancing the rehabilitation experience.

2. Problem description

As part of the Roboasset project, a platform with a robotic arm has been integrated with a VR game that simulates a ski track. The objective is to create an immersive simulation in which the patient can perform rehabilitation movements while skiing (Figure 1).

The robot's control is designed on ROS, allowing communication with Unity, the game development engine used. This communication is carried out via a TCP port created from the robot's side, through which the pose of the robot's end effector, acting as a ski pole, is sent. For the user to grasp the pole correctly, it is necessary to achieve real-virtual equivalence.

Figure 1: Ski virtualization demo with robotic interaction.

The initial setup includes two AprilTag markers, known for their superior orientation detection (Kalaitzakis et al., 2021). One marker (*realHeadMarker*) is placed on the front of the VR headset and another (*tableMarker*) on a static position of the robot's platform. A Realsense D435i camera (*cam*), mounted on a tripod at a height of 1.2 meters, captures the entire environment from a position approximately 2 meters in front of the platform, ensuring optimal framing of the markers within its field of view.

In the virtual environment, a twin platform with the same dimensions needs to be created, including the static marker and the robot's base. This replica is essential for positioning the pole with the robot effector's poses, which are expressed in the base's coordinate system. In addition, due to software limitations, we cannot directly overwrite the virtual camera to reposition the user's point of view; rather, we must manipulate the container, which serves as the reference system for it.

The pre-simulation calibration lasts several seconds and involves a series of calculations to obtain the transformation matrices, represented as a transformations tree in Figure 2. These calculations are based on the assumption that the real and virtual platform markers are coincident. We will now go through the steps to find the transformation matrix.

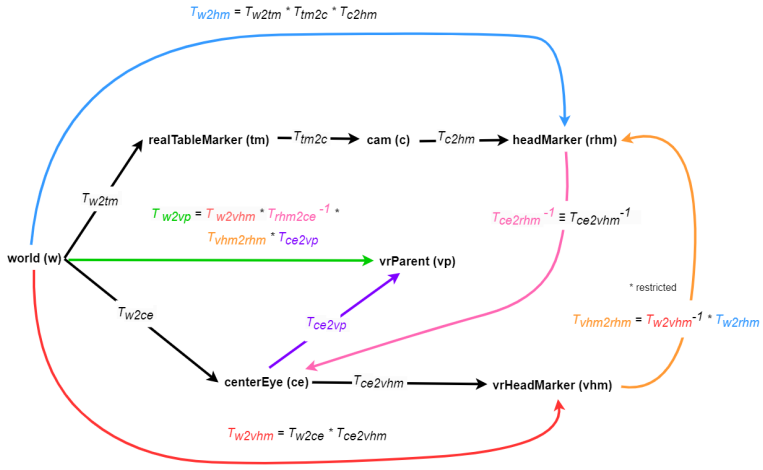

Figure 2: Transformation tree.

The first branch of the transformation tree, the real one, extracts the matrix from *world* to *realHeadMarker* (T_{w2rhm})

coordinate systems, represented in the Figure 2 in blue. To achieve this, several intermediate transformations are passed through: from *world* to *tableMarker* (T_{w2tm}), from *tableMarker* to *cam* (T_{tm2c}), and from *cam* to *realHeadMarker* (T_{c2rhm}). The target matrix is calculated as:

$$T_{\text{w2rhm}} = T_{\text{w2tm}} \cdot T_{\text{tm2c}} \cdot T_{\text{c2rhm}}$$

In the virtual environment branch, three systems associated with the POV are defined: the estimation of the virtual marker, *vrHeadMarker* (vhm); the virtual POV camera, *centerEye* (ce); and the container that allows repositioning it, *vrParent* (vp). The transformation T_{w2vhm}, represented in Figure 2 in red, is calculated as:

$$T_{\text{w2vhm}} = T_{\text{w2ce}} \cdot T_{\text{ce2vhm}}$$

The transformation T_{ce2vhm} is an estimated translation of 3 cm vertically and 7 cm backward, given that the center of the eyes is located at that distance from the front of the headset, as represented in Figure 3.

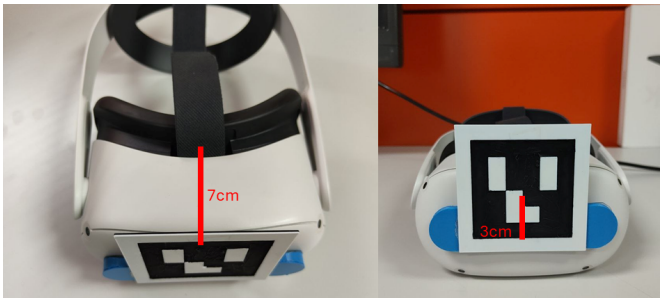

Figure 3: Marker position from virtual center eye.

To obtain the transformation between *vrHeadMarker* and *realHeadMarker*, the previously calculated transforms are multiplied. The resultant has to be restricted to vertical rotation only, since the slightest effect in the other directions would tilt the ground reference, causing vertigo in the user during the immersive session. This is expressed as follows:

$$T_{\text{vhm2rhm}} = Restrict_{\text{y}}(T^{-1}_{\text{vhm2w}} \cdot T_{\text{w2rhm}})$$

Finally, the transformation that repositions the POV through *vrParent*, represented in Figure 2 in green, is calculated as:

$$T_{\text{w2vp}} = T_{\text{w2vhm}} \cdot T_{\text{vhm2rhm}} \cdot T_{\text{rhm2ce}} \cdot T_{\text{ce2vp}}$$

We preform an average of the transformation, $T^{-1}_{\text{vhm2w}} \cdot T_{\text{w2rhm}}$ over N measurements, to refine the calibration. The averaging of these transformations and the inner workings of the *Restrict* function will be described in the nex section.

3. Quaternion Geometry

To solve the problem of restricting rotations, we need to first define it formally. We will use unit quaternions to represent the rotation group $SO(3)$. Geometrically, unit quaternions (referred to simply as quaternions) form the surface of a 4-dimensional sphere of radius 1, with components x, y, z, and w. The 4-sphere is a double cover of the rotation group, as both a quaternion q and its negative $-q$ represent the same rotation.

The surface of an n-sphere is locally similar to $\mathbb{R}^n$, making the sphere a smooth manifold. As a smooth manifold, we can attach a tangent plane at any point, representing directions and velocities at that point. By following a direction and rolling the tangent plane, we define a curve along the sphere, as shown in Figure 4. On the sphere, curves following a constant direction are geodesics, which are the shortest paths between any two points on the curve. These are analogous to straight lines in $\mathbb{R}^n$ space

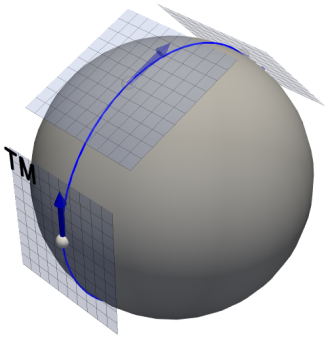

Figure 4: Obtaining the geodesic by rolling the tangent plane.

Isolating the part of the rotation that corresponds to a rotation purely in y involves finding the closest point on the geodesic generated by the direction y, as shown in Figure 5. These lines intersect at a 90-degree angle, analogous to a projection onto an axis in $\mathbb{R}^n$.

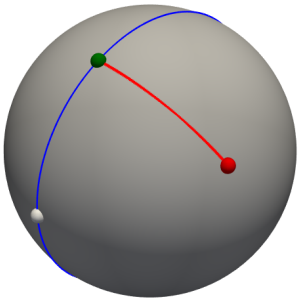

Figure 5: Geodesic projection of the rotation

.

Due to the geometry of the sphere, finding this projec-

tion is straightforward. All geodesics are arcs of great circles, which have the same center and radius as the sphere. We are interested in the one perpendicular to the YW-plane. Then we simply have to project the point, as represented in Figure 6.

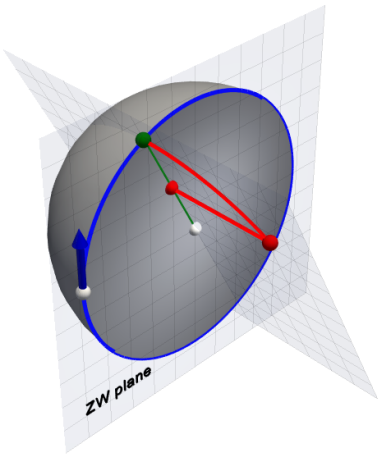

Figure 6: Projecting to the YW plane, then to the sphere.

To project the point, a 4-dimensional projection matrix is used. This can be easily defined as:

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

This makes the $Restrict_y(q)$ function for any quaternion the following:

$$Restrict_y(q) = \frac{Pq}{\|Pq\|}. \tag{2}$$

Here, P is the projection matrix to the ZW plane, q is the original quaternion, and q_y is the rotation restricted to the y component. The rotation $Restrict_y(q)$ is now guaranteed to only rotate the coordinate system of the VR world in around its vertical axis y. Preventing inducing any kind of tilt ind the horizontal floor plane.

There are different methods to take the average of a set of quaternions, with different pros and cons. Minimizing the geodesic distance of the average quaternion to all other quaternions would be the stricter method, but is computationally expensive. Since our quaternions obtained from measurements are expected to be similar, we can simply take their average as 4-vectors and then normalize the result so that it lies on the surface of the 2-sphere. This assumes that over small distances, the surface of the sphere is very similar to a plane.

A final issue to take into account when taking the average is to ensure that all quaternions are in the same "side" of the sphere. Since, as we first described, q and $-q$ represent the same rotation. To do this, we save the first quaternion q_0 and perform the vector dot product with the rest of quaternions.

$$\mu(Q) = \frac{\sum_{q_i}^{Q} \sigma(q_i)}{\| \sum_{q_i}^{Q} \sigma(q_i) \|}, \tag{3}$$

where Q is the set of quaternions to average, and σ is the sign function defined as follows using the first received quaternion q_0,

$$\sigma = \begin{cases} q & \text{for } q \cdot q_0 >= 0 \\ -q & \text{for } q \cdot q_0 < 0 \end{cases} \tag{4}$$

Taking the averages of the measured positions is simply taking the average of vectors. Then fromthe average of positions $\mu(p)$ and the average of rotations $\mu(Q)$ we can create an average transformation matrix

$$\mu(T) = \begin{bmatrix} R & \mu(p) \\ 0 & 1 \end{bmatrix} \tag{5}$$

With R

$$R = \begin{bmatrix} 1 - 2(q_j^2 + q_k^2) & 2(q_i q_j - q_k q_w) & 2(q_i q_k + q_j q_w) \\ 2(q_i q_j + q_k q_w) & 1 - 2(q_i^2 + q_k^2) & 2(q_j q_k - q_i q_w) \\ 2(q_i q_k - q_j q_w) & 2(q_j q_k + q_i q_w) & 1 - 2(q_i^2 + q_j^2) \end{bmatrix} \tag{6}$$

where q_x, q_y, q_z, and q_w are the components of the average quaternion $\mu(Q)$.

4. Experimetns and Results

The setup consists of a robot on a fixed base with an ArUco positioned in the corner, a Meta Quest 2 Headset with a second ArUco positioned in its front.

Before the real test, we perform a preliminary synthetic test to measure the accuracy of the system. We place the arucos on a table with known distances between them $\delta_x = 0.2m, \delta_y = 0.2m, \delta_x = 0.105m$ and no difference in rotation, as shown in Figure 7

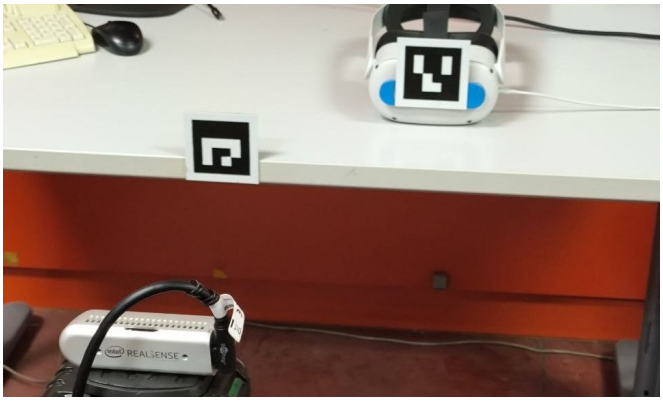

Figure 7: Setup to measure accuracy in position and rotation error.

We found the error in position to be under 5mm and the error in y-axis rotation under 1.5 degrees.

After the calibration, the results were accurate, allowing the user to verify the alignment with virtualized real elements

such as the table or the robot's base. This precise calibration ensures that the virtual representations correspond correctly to their real-world counterparts.

Finally, we perform a qualitative test where a subject is placed next to the robot base, Figure 8. The calibration is then performed, allowing the subject to touch the virtual robot base and assess its correct positioning. As seen in the last image of the figure, the subject's hand correctly sits on the virtual table in the same position and orientation as in the real world.

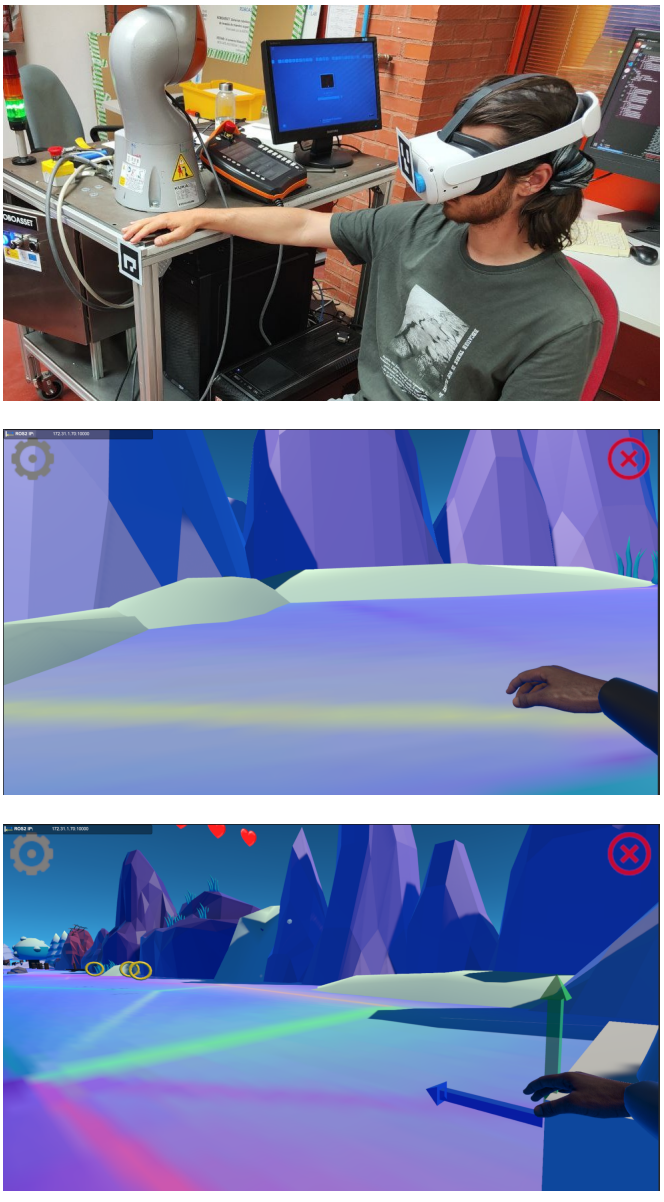

Figure 8: Comparison User point of view before and after calibration.

From an external point of view the real world calibration is seen more clearly. In Figure 9 we see the users avatar starting in a random position relative to the virtual table. And after applying the calibration, its virtual positioning is identical to the real world image sen in Figure 8

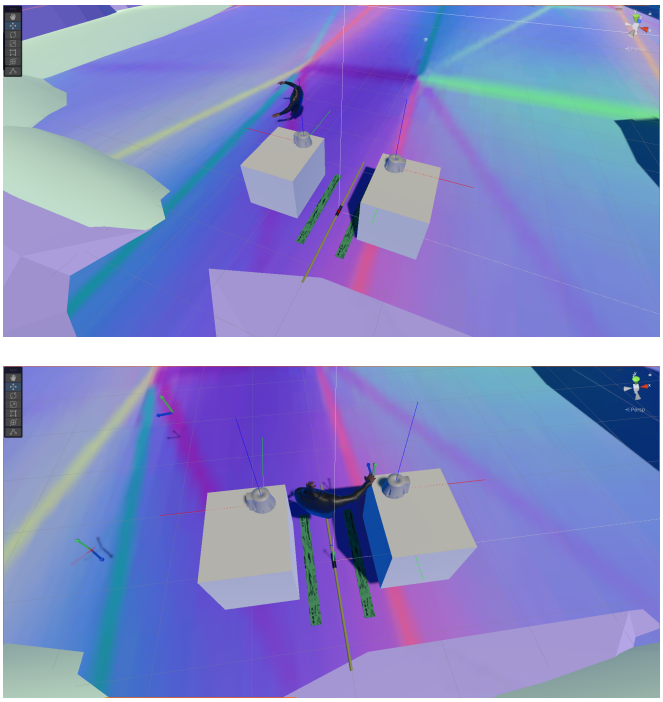

Figure 9: Comparison of VR world positioning before (top) and after calibration (bottom).

The vertical restriction applied during calibration was crucial; without it, rotations in other directions would create a mismatch between the virtual environment and the user's physical perception, leading to sensations of vertigo and instability. This mismatch occurs because the user relies on tactile feedback from the ground, and any discrepancy between the virtual and real rotations disrupts this alignment. By applying the vertical restriction, we ensure a stable and immersive experience that aligns with the user's physical sensations and movements.

Agradecimientos

Estos resultados son posibles gracias a la financiación del Ministerio de Economía y Competitividad español como parte del proyecto: "'ROBOASSET: Intelligent robotic systems for assessment and rehabilitation in upper limb therapies" (PID2020-113508RB-I00), financiado por AEI/10.13039/501100011033, y el proyecto: "iREHAB: AI-powered Robotic Personalized Rehabilitation", (ISCIII-AES-2022/003041), financiado por el Instituto de Salud Carlos III (ISCIII) y cofinanciado por la UE. Además esta publicación es parte del proyecto de I+D+i IROPER PLEC2021-007819 financiado por MCIN/AEI/10.13039/501100011033 y por la Unión Europea "NextGenerationEU/PRTR

References

Bardorfer, A., Munih, M., Zupan, A., Primozic, A., Sep. 2001. Upper limb motion analysis using haptic interface. IEEE/ASME Transactions on Mechatronics 6 (3), 253–260.
DOI: 10.1109/3516.951363

Barnett-Cowan, M., Barnett-Cowan, M., Harris, L. R., Harris, L. R., 2008. Perceived self-orientation in allocentric and egocentric space: effects of visual and physical tilt on saccadic and tactile measures. Brain Research.
DOI: 10.1016/j.brainres.2008.07.075

Ceyte, H., Ceyte, H., Cian, C., Cian, C., Trousselard, M., Trousselard, M., Barraud, P., Barraud, P.-A., 2009. Influence of perceived egocentric coordinates on the subjective visual vertical. Neuroscience Letters.
DOI: 10.1016/j.neulet.2009.06.048

Geest, S. D., Geest, S. D., Sabaté, E., Sabaté, E., Sabaté, E., 2003. Adherence to long-term therapies: Evidence for action:. European Journal of Cardiovascular Nursing.
DOI: 10.1016/s1474-5151(03)00091-4

Jiang, W., Jiang, W., Jiang, W., Yuan, X.-Z., Yuan, X., Yin, C., Yin, C., Wei, K., Wei, K., 2018. Visuomotor learning is dependent on direction-specific error saliency. Journal of Neurophysiology.
DOI: 10.1152/jn.00787.2017

Kalaitzakis, M., Kalaitzakis, M., Cain, B., Cain, B., Carroll, S., Carroll, S., Ambrosi, A., Ambrosi, A., Whitehead, C., Whitehead, C., Vitzilaios, N., Vitzilaios, N. I., 2021. Fiducial markers for pose estimation. Journal of Intelligent and Robotic Systems.
DOI: 10.1007/s10846-020-01307-9

Laver, K., Laver, K., George, S., George, S., Thomas, S., Thomas, S., Deutsch, J. E., Deutsch, J. E., Crotty, M., Crotty, M., 2015. Virtual reality for stroke rehabilitation. Cochrane Database of Systematic Reviews.
DOI: 10.1002/14651858.cd008349.pub3

Lewis, G., Rosie, J., 04 2012. Virtual reality games for movement rehabilitation in neurological conditions: How do we meet the need and expectations of the users? Disability and rehabilitation 34, 1880–6.
DOI: 10.3109/09638288.2012.670036

Michiels, N., Jorissen, L., Put, J., Liesenborgs, J., Vandebroeck, I., Joris, E., Reeth, F., 2024. Tracking and co-location of global point clouds for large-area indoor environments. Virtual Reality.
DOI: 10.1007/s10055-024-01004-0

Reimer, D., Reimer, D., Podkosova, I., Podkosova, I., Scherzer, D., Scherzer, D., Kaufmann, H., Kaufmann, H., 2021. Colocation for slam-tracked vr headsets with hand tracking. The first computers.
DOI: 10.3390/computers10050058

Steiner, B., Elgert, L., Saalfeld, B., Wolf, K.-H., Aug 2020. Gamification in rehabilitation of patients with musculoskeletal diseases of the shoulder: Scoping review. JMIR Serious Games 8 (3), e19914.
URL: http://games.jmir.org/2020/3/e19914/
DOI: 10.2196/19914

Veerbeek, J. M., Veerbeek, J. M., Langbroek-Amersfoort, A. C., Langbroek-Amersfoort, A. C., van Wegen, E. E. H., van Wegen, E. E. H., Meskers, C. G., Meskers, C. G. M., Kwakkel, G., Kwakkel, G., 2017. Effects of robot-assisted therapy for the upper limb after stroke. Neurorehabilitation and Neural Repair.
DOI: 10.1177/1545968316666957

Vogels, R., Vogels, R., Vogels, R., Orban, G. A., Orban, G. A., 1985. The effect of practice on the oblique effect in line orientation judgments. Vision Research.
DOI: 10.1016/0042-6989(85)90140-3