



Development of an Adaptable 3D Rigged Model for the Design of Bespoke Devices for Upper Limb Rehabilitation

Beatrice Aruanno¹ , Sandro Barone² , Paolo Neri³ , Alessandro Paoli⁴  and Francesco Tamburrino⁵ 

¹University of Pisa, Department of Civil and Industrial Engineering beatrice.aruanno@ing.unipi.it

²University of Pisa, Department of Civil and Industrial Engineering sandro.barone@unipi.it

³University of Pisa, Department of Civil and Industrial Engineering paolo.neri@unipi.it

⁴University of Pisa, Department of Civil and Industrial Engineering alessandro.paoli@unipi.it

⁵University of Pisa, Department of Civil and Industrial Engineering francesco.tamburrino@unipi.it

Corresponding author: Beatrice Aruanno, beatrice.aruanno@ing.unipi.it

Abstract. The human hand is a versatile and complex body part. It permits difficult movements with various degrees of precision and force. Several causes can lead to upper limb damage, including musculoskeletal disorders and diseases like stroke. The impairment can affect daily living activities. Patients usually undergo rehabilitation therapy with medical personnel for a long time after the traumatic event. In most cases, they use off-the-shelf medical devices. However, the shape of the upper limbs can differ a lot among people. A bespoke rehabilitative device could provide better comfort and usability, but the design process can be challenging. This work aims to present a digital workflow to generate a 3D virtual reconstruction of the patient's upper limb structure, to be used in the device design. Starting from a 3D scan acquisition of the patient's upper limb, the algorithm allows the creation of a polygonal mesh of the arm and the hand by a semi-automatic procedure. The algorithm uses neural networks' capability to automatically detect the upper limb's landmarks to localize the joints' coordinates. The joints' positions can be used to build a virtual skeleton for a 3D model of a human arm. The mesh of the model is subsequently wrapped around the scan of the real arm. The output consists in the 3D rigged model of the patient's upper limb with a manifold mesh that can be deformed using its virtual skeleton. The results have been assessed with patients who had sports injuries or strokes. The 3D deviations between the scan acquisition of the arm and the resulting model have been evaluated.

Keywords: 3D virtual reconstruction, upper limb rigged model, 3D scan, automatic landmarks detection, neural network

DOI: <https://doi.org/10.14733/cadaps.2023.S6.75-86>

1 INTRODUCTION

The design process of a bespoke rehabilitative device for upper limbs can be challenging. The human hand is a complex body part that can also perform complex functions. It permits difficult movements with various degrees of precision and force. There are numerous robotic devices and exoskeletons in the literature that aim to assist patients who suffer from hand impairments caused by traumatic events or diseases [5], [2]. Stroke, musculoskeletal disorders, or spinal cord injury can affect the execution of daily living activities, with the resulting loss of independence and quality of life. Patients undergo rehabilitation therapy with medical personnel for a long time after the traumatic event, and only a limited number of solutions involve telerehabilitation systems [15], [1]. In most cases, they use off-the-shelf medical devices. However, the shape of the upper limbs can significantly differ among people. A bespoke rehabilitative device could provide better comfort and usability.

One of the first steps in the custom design process involves the acquisition of patient-specific data. In most cases, the process is still manually executed by medical personnel or therapists with rulers and goniometers. However, this process is labor-intensive and highly dependent on the ability of the technician. Furthermore, the data available are limited only to those manually recorded. From the perspective of a digital workflow, in this work, a 3D scanner was used to acquire the whole surface of the upper limb to be used as input for the design phase.

Once the output of the 3D scan of a body part has been processed, the mesh may present some artifacts (e.g. holes, spikes) largely due to self-occlusions and undercuts or small uncontrolled movements during the acquisition. Obtaining a manifold mesh that can be used in the parametric design of the bespoke rehabilitative device can be challenging.

This research, which was developed in the framework of the PRIME-VR2 European Project, aims to present an algorithm to generate a custom 3D rigged model of the upper limb of a patient starting from a 3D scanning acquisition, moving through the point clouds semi-automatic elaboration using machine learning.

2 3D SCANNING OF UPPER LIMBS

The reconstruction of a 3D mesh of human body parts can be difficult due to undercuts and the occlusions they generate. In particular, the 3D scanning protocol for upper limbs requires multiple frames from different viewpoints to obtain a complete acquisition of the body part. Furthermore, it must be taken into account that the person should stay still for the whole duration of the scanning time, otherwise it would not be possible to correctly align the single frames and the result could not faithfully represent reality. This aspect becomes even more relevant when the person to scan is a patient, who can have higher difficulties in maintaining the pose for several minutes. For this reason, the scanning procedure should be completed in the range of one minute and must be sufficiently robust to acquire the whole surface of the upper limb without erroneous artifacts.

To overcome this aspect, a specific 3D scanner was developed [14]. It uses three low-cost D415 Intel® RealSense™ depth cameras fixed on a common rigid frame [9]. The combination of multiple independent sensors allows to widen the field of view of each frame, therefore it reduces the total number of acquisitions necessary to complete the scanning procedure. Furthermore, a slightly different orientation among the cameras allows the acquisition of some hidden areas that otherwise would require a specific acquisition. For example, a typical situation of undercut with upper limbs is the area on the back of the thumb. To combine these two advantages and also to obtain a lightweight and versatile device, the cameras mounted on a ring are angled in both axial and radial directions. The final layout of the prototype can be seen in Figure 1(a).

Each camera is equipped with a high-resolution RGB sensor (1920x1080 pixels) and a depth camera (1280x720 pixels), based on active infrared stereoscopic technology. Thus, it can acquire a color picture and a point cloud of the scene for each frame. Considering that the three sensors are rigidly connected to the others by the circular frame, it is possible to register the single frames of the cameras to obtain a unique acquisition. A preliminary calibration procedure is used to determine

the roto translation matrices that refer to the 3D position and orientation of each sensor with respect to the others.

The global 3D scan of a patient's arm requires several consecutive acquisitions. The developed device should be handheld by the therapist and moved around the upper limb. After numerous tests performed by researchers and therapists, it emerges that approximately 6 acquisitions are sufficient to reconstruct the full 360° surface of an arm. Figure 1(b) shows an example of a one-frame acquisition.

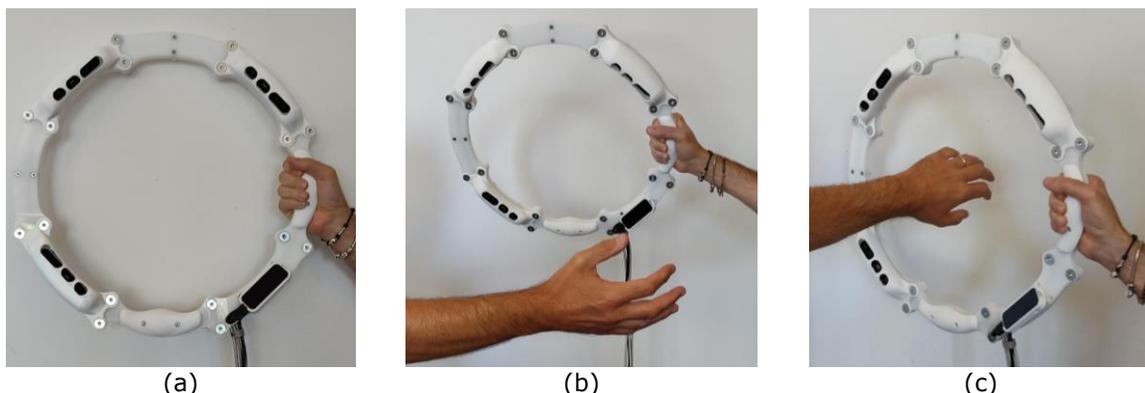


Figure 1: The prototype of the developed 3D scanner (a) and two examples of the system used to acquire the upper limb surface of a person (b, c).

3 SEMI-AUTOMATIC KEYPOINTS DETECTION

Once the 3D scanning process is complete, the result is a set of triplets of point clouds and RGB images from each acquisition. The following step consists in point cloud elaboration and alignment. The conventional workflow involves the pairwise registration of cleaned and denoised clouds with the so-called 3-2-1 method [16]. It involves the manual selection of at least three corresponding points on adjacent point clouds. A detailed explanation of the full procedure can be found in a previous work [14] and it is outside of the aim of this paper. However, this procedure is labor intensive and requires a relevant amount of time. Furthermore, it is highly dependent on the user's experience.

To accelerate this step, an algorithm that automatically detects corresponding notable points on the point clouds was used. The algorithm exploits the potentiality of neural networks (NN) and machine learning (ML) to infer the location of these key-points on RGB images acquired by the sensors during the 3D scanning. In literature, several NNs exist, and the "MediaPipe Hand" by Google, which infers 21 landmarks on the hand from a single image, was selected. One key-point is located on the wrist, 15 on the fingers' joints location, and 5 at the tip of the fingers [13]. The piece of code is written in Python and can be easily combined with the point clouds elaboration process. The output consists of an ordered list with 21 bidimensional coordinates of the key points (KPs) on the image, plus the depth coordinates estimated by the ML model and an overall score that indicates the detection confidence value.

The KPs can be reprojected on the 3D scan mesh and used to accelerate the 3-2-1 method. Further details about the algorithm can be found in a previous paper [14]. However, it also allows to obtain a list of 3D coordinates on the mesh at the joints' position of the 3D scan. The KPs are stored always in the same order, and if any point is not detected it is replaced with a *not-a-number* value in the list. This important aspect allows a one-to-one correspondence between the list item (and coordinates) and the landmark of the hand. During the point cloud elaboration, two others points are manually selected on the mesh to identify the elbow and the beginning of the arm. Then,

the 23 points with the 3D coordinates are exported and stored in a text file. These points will be used as destinations for the skeleton's bones of the rigged model that will be described in the next paragraph. At the end of the point clouds elaboration process, the output is a 3D mesh of the upper limb of a patient stored as a surface file (e.g., *step* or *ply*).

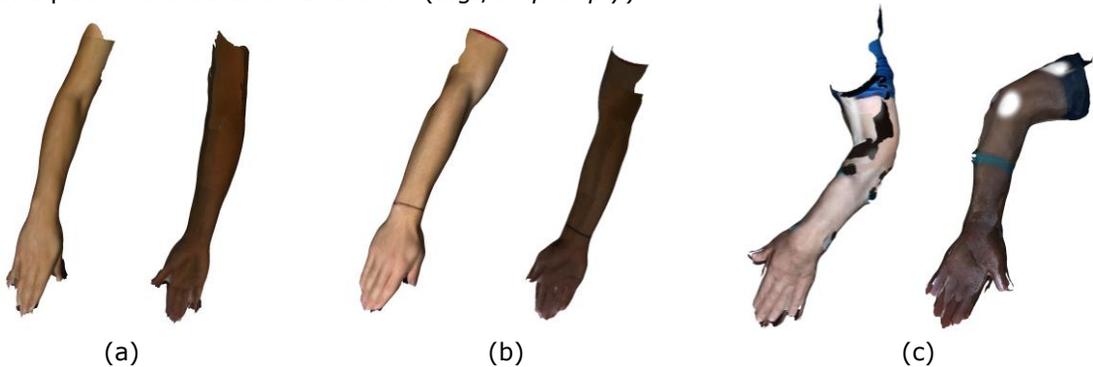


Figure 2: Example of three acquisitions of upper limbs of different persons. Figures represent the upper and lower view of the patient's arm acquired by the 3D scanner. Two persons had a sports injury (a, b) and one had a stroke (c).

The quality of the resulting mesh depends on different factors including the ability of the inexperienced therapist to use the scanner, the health status of the patients, or their capability to maintain the arm still for the whole duration of the scanning process. Several tests with therapists and their patients who had a sports injury or stroke have been performed. As can be seen in Figure 2 (a, b), some cases result in a complete surface of the arm and some small holes mainly on the fingertips. But there are also cases where the mesh presents a higher number of holes spread over the arm (Figure 2(c)). In all these latter situations it is difficult to obtain a manifold mesh without holes on the arm's surface that can be directly used to design a bespoke controller.

4 ADAPTABLE 3D RIGGED MODEL

An additional procedure is then required when it is not possible to generate a manifold mesh directly from the scanning process. This paper presents the workflow to generate an adaptable 3D rigged model fitted to the real 3D scan acquisition.

The term rigged model identifies a 3D mesh that can be deformed and animated using a hierarchy of interconnected bones that constitute the skeleton of the model. Every mesh can be rigged: humanoid characters, objects, animals, or even plants. The skeletons do not require to correspond to any anatomical feature of the real world. Rigging is a technique widely used in computer graphics and character animation. The most used approach is based on the skeletal animation introduced in 1988 [11]. Various computational methods exist in the literature for automatic rig [4], [7], but it is still commonly performed by hand. Figure 3 illustrates an example of the workflow for a human arm. It mainly consists of two steps. Once the 3D mesh of the subject is completed (a), a series of interconnected bones is created. Each bone can be moved, rotated, and scaled. One advantage of rigging is that the assembly of bones forms a hierarchical skeleton (or rig), with parent and child relationships (b). It means that a child bone's transformation is influenced by its own transformation but also by the parent's ones. In this example, if the wrist bone is rotated along one axis, all the fingers-bones will follow the movement. However, till this moment, the mesh is not connected to the skeleton. To obtain this goal a second step is necessary, called skinning. This process binds the vertices of the mesh to the bones. Each part of the mesh will be influenced by one or more bones of the skeleton. How each bone affects the deformation of some specific mesh nodes is defined using skin weights. There are numerous techniques to generate this weight map

automatically [6], [8]. The most common procedure computes the distance between each vertex and the selected joint in a delimited area, the lower the distance the higher will be the weight that the bone exerts on that vertex's deformation. However, this process does not always generate the desired results [10], and skinning a new mesh requires a manual refinement of the weights with specific weight-paint tools of modeling software (e.g., Blender). Figure 3(c) illustrates, with a color map, the skin weights for the lower arm bone. The weights are ranged from zero (blue, not influenced) to one (red, highly influenced). It is important to highlight that even though the hand is not directly deformed by the lower arm, the wrist and the fingers' bones will follow the transformation due to the parent-child relationship. Figure 3(d) shows an example of how the mesh can be deformed using the skeleton and the skin weights.

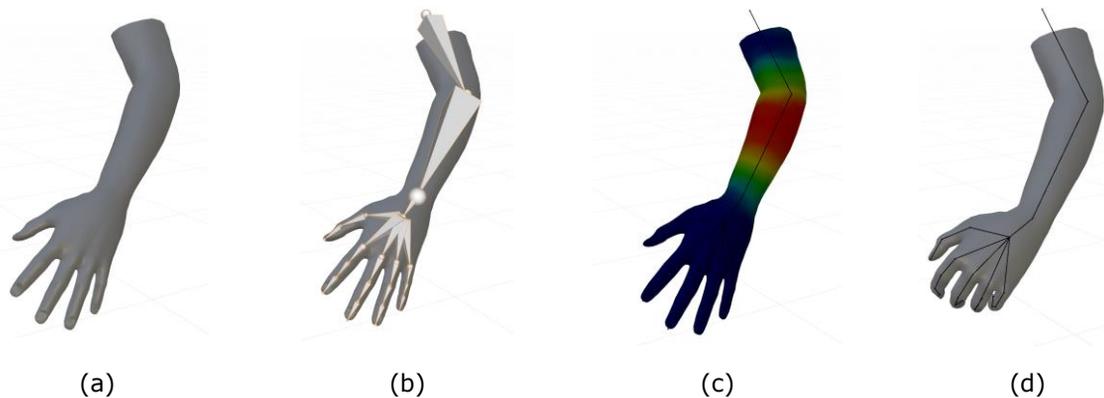


Figure 3: Graphical description of the rigging procedure. Starting from a mesh (a), the skeleton is created (b). Then, the weights of each bone are assigned to the mesh vertices, here are represented the weights for the lower arm (c). Once these steps are completed, the model is rigged and can be deformed using the skeleton (d).

In this work, the potentiality of a deformable rigged model is exploited to correct and complete the 3D scan acquisition of the upper limb that could present numerous artifacts, such as holes, internal faces, non-manifold mesh, disconnected components, etc.

As detailed in the previous paragraph, generating a rigged model from scratch can be a labor-intensive process and requires a notable amount of time. For this reason, since the target meshes acquired with the 3D scanner are upper limbs of patients, a generic rigged model of a human arm could be used. However, people are very different one from each other, so we decided to use MakeHuman™ open-source software [12] to generate the generic humanoid character and combine it with the modeling software Blender. With the MakeHuman™ plugin for Blender (MPFB2.0), it is possible to generate directly inside the modeling software the mesh of a virtual human and personalize their anatomical proportions using a graphical user interface. As detailed in Figure 4(a), the scan is imported into Blender and a generic virtual human is created in the same scene.

Blender allows altering the mesh in a non-destructive way using a set of automatic operations named "modifiers". Several modifiers can be added to an object and the order in the stack defines how the mesh is changed. This works in real-time, but to make the changes permanent (e.g., before exporting the mesh), all the modifiers must be applied [3]. Considering that only the portion of the model corresponding to the upper limb is relevant to the scope, this part is isolated (Figure 4(b)). The needed vertices of the arm are selected on the model and are stored in a new vertex group. Then, by adding a Mask modifier to the stack of the model, it is possible to hide all other vertices and faces not included in the desired group.

The MPFB2.0 plugin integrates a graphical user interface that allows the modification of the humanoid in all its aspects, including the proportions of the body and the size of each part. For

instance, as shown in Figure 4(c), it is possible to tune the length and the depth of the lower arm, the volume of fat in this part of the body, or the width of the hand and the size of the fingers (not shown in this picture). In this work, we aligned the 3D scan and the model in different views and roughly fit the length and width of the mesh. We notice that this manual step improved the final result in terms of the overall regularity of the output mesh.

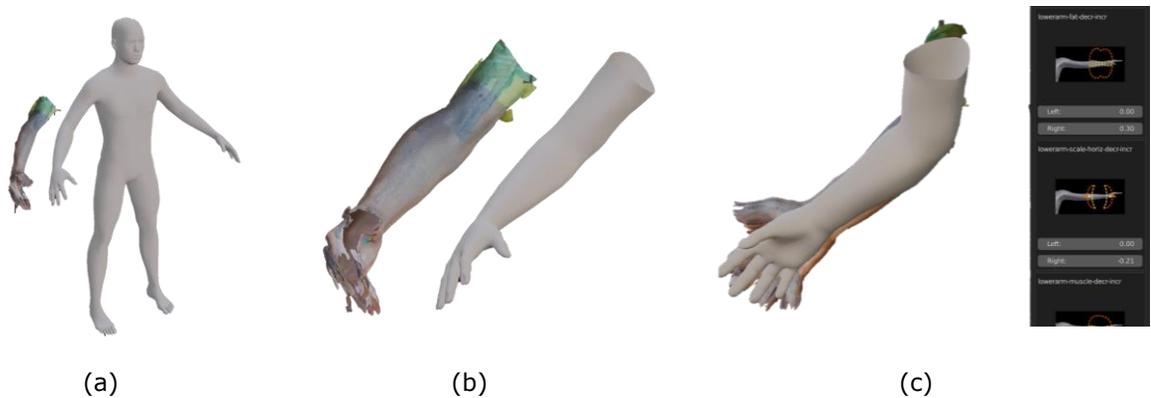


Figure 4: Workflow to generate a custom arm combining Blender with the MakeHuman™ plugin. The mesh of the scan and a default human model are imported into a common scene (a). Only the vertices of the corresponding arm are kept on the model (b). The mesh is approximately aligned to the model in different views, and using the sliders of the plugin is possible to roughly fit the length and distances of the model's arm and hand (c).

Once the global shape of the model has been defined using the graphical user interface of MPFB2.0, the plugin allows generating the complete rig. It directly creates the skeleton and the weights for each bone of the upper limb. As can be seen in Figure 5(a2), the model has all the necessary bones (illustrated as octahedrons) and joints (illustrated as spheres) to be deformed as the pose of the 3D scan of the patient.

Python scripts can extend the functionalities of Blender and can be used to automatize numerous aspects of the modeling software. In this case, we used a script to move the joints of the model's bones to the coordinates of the 23 KPs reprojected on the 3D scan. Only for visualization purposes, a skeleton has been created (and represented as sticks) at the KPs' coordinates. Figure 5(a1) shows the position of the KPs connected with bones on the mesh. Using a Python script, it was possible to move the rig of the model at the desired coordinates and roughly align the two meshes (Figure 5(b)).

To improve the overlap between the two arms' surfaces, the "Shrinkwrap" modifier provided by Blender is used. It stretches the vertices of a mesh toward the surface of another object. Different methods are available in the modifier to detect each nearest point for each vertex of the modified mesh. In this case, each vertex is moved to the closest point detected on the surface of the 3D scan. Figure 5(c) shows an example of the result after the Shrinkwrap modifier has been applied. On the final mesh, a Subdivision Surface Modifier is applied to reduce possible spikes and give a smooth appearance.

A separate consideration must be done for the hand. Usually, the areas between the fingers and the fingertips are the most difficult to acquire due to self-occlusions or small movements during the scanning procedure. These spots can present more holes or artifacts compared to the arm's surface. Deforming the hand as the rest of the mesh can generate unexpected behaviors. To reduce this event, it is possible to assign a lower weight to the hand or in the areas where the scan can present greater problems.

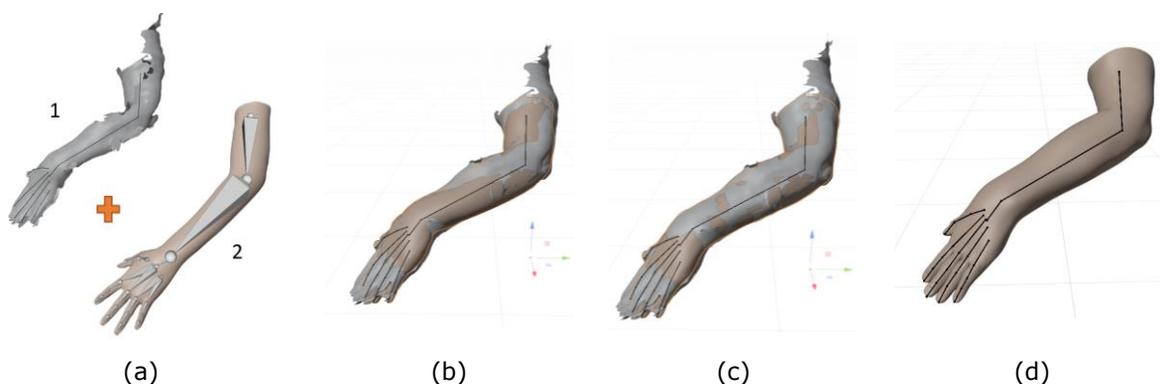


Figure 5: The patient's 3D scan and the rigged model of the arm in Blender, and the skeleton of the scan is created (a). The two models are roughly aligned using the bones' positions (b). The fine alignment is obtained using the Shrinkwrap algorithm (c), to obtain the final rigged model (d).

The modifier allows the definition of a specific group of vertices of the mesh to be deformed. Similarly to the skin weight used for rigging, a higher value was assigned to the weights on the arm and a lower value was used for the hand with a smooth distribution at the interface. Figure 6(a) illustrates the step, while Figures 6(b, c) show the comparison between the two versions of the model without (b) and with (c) the Shrinkwrap modifier.

The result of this procedure is a manifold mesh without holes that can be used as a starting point in a digital procedure for a bespoke upper limb rehabilitation controller.

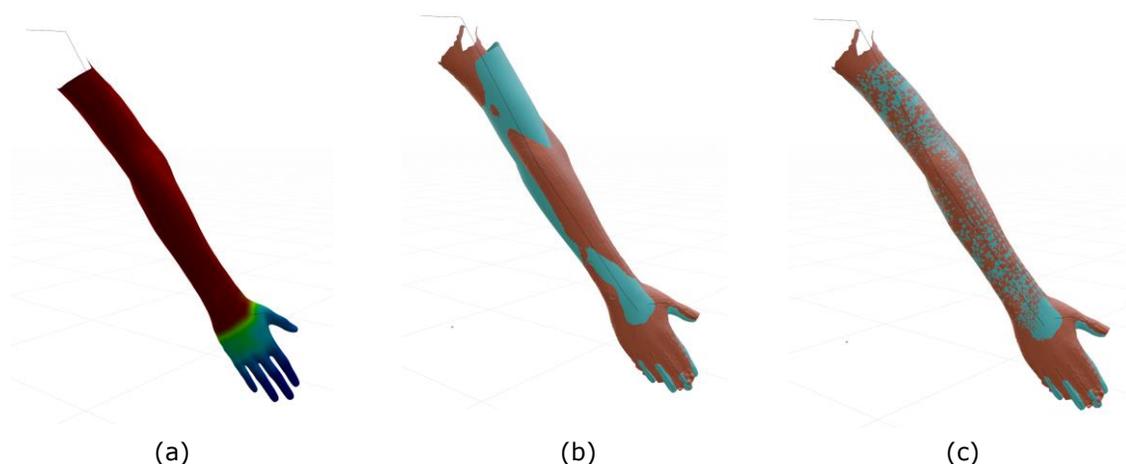


Figure 6: The Shrinkwrap modifier workflow. The weight distribution on model (a) defines how much the vertices of the model will deform towards the target mesh' points. The model (blue) and the 3D scan (orange) before (b) and after the modifier (c).

5 RESULTS

This procedure has been applied to five different scans of the upper limbs of healthy subjects, patients who had sports injuries, and stroke patients to broaden the number and specificity of cases. The model and the 3D scan are aligned after the rigging process. So, the 3D deviation map was

computed, to analyze how much the model reflects the 3D scan and which areas present higher discrepancies

Figure 7 shows an example of a 3D deviation analysis of patients who had sports injuries (a) and stroke (b). The colormap scale spans between $\pm 5\text{mm}$, but from the histograms emerges that the majority of values are approximately in the range $+2.5\text{mm}/-1\text{mm}$ with higher values in the stroke patient's case. The highest deviations are concentrated in the area of the hand. There are some grey areas. This is due to the presence of holes in the scan surface (which are not present in the final manifold mesh), thus the algorithm could not compute the deviations in those points.

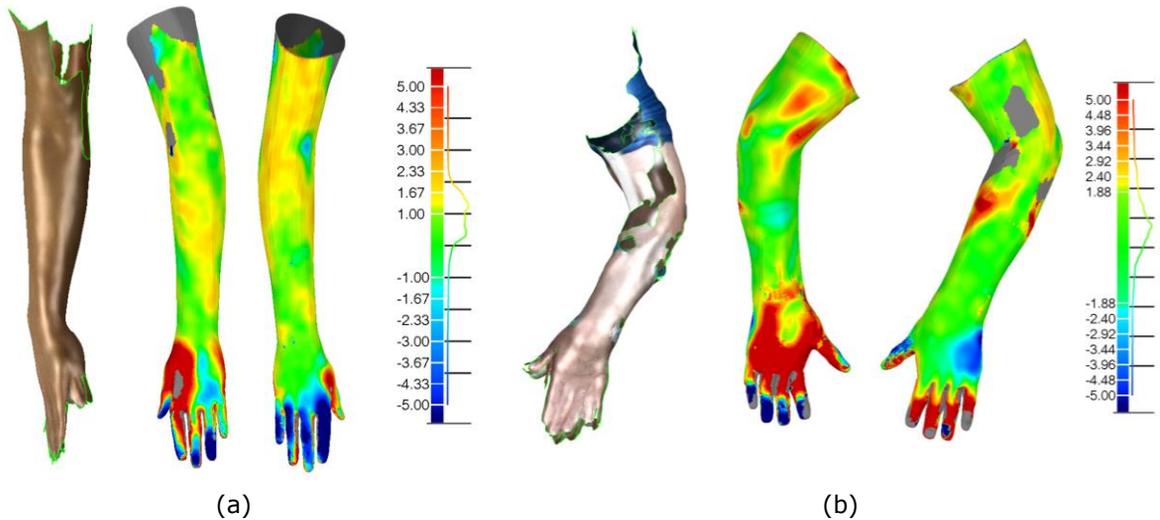


Figure 7: Two examples of 3D deviation analysis between the rigged model and the 3D scan of a patient with sports injury (a) and a patient who had a stroke (b).

Table 1 reports the overall average values and standard deviation of the five cases examined. They are coupled with a histogram that shows the distribution of the 3D deviations. The comparisons have been made by setting the rigged model as a reference and the 3D scan acquisition as a tester. Positive values (towards the red color) indicate that the model surface is outside the 3D scan. On the other hand, negative values (toward the blue) indicate points that lay inside the scan.

Scan	Avg [mm]	St Dev [mm]	Histogram
1	0.85 (1.41/-1.59)	2.22	
2	0.89 (1.18/-1.14)	1.49	
3	1.34 (1.46/-0.54)	1.75	

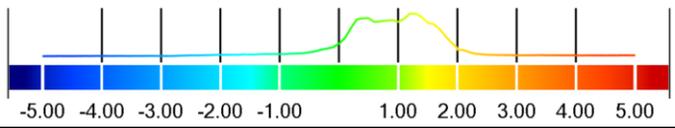
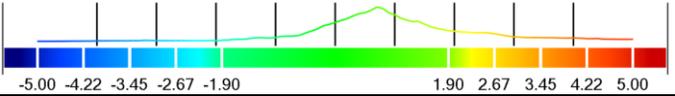
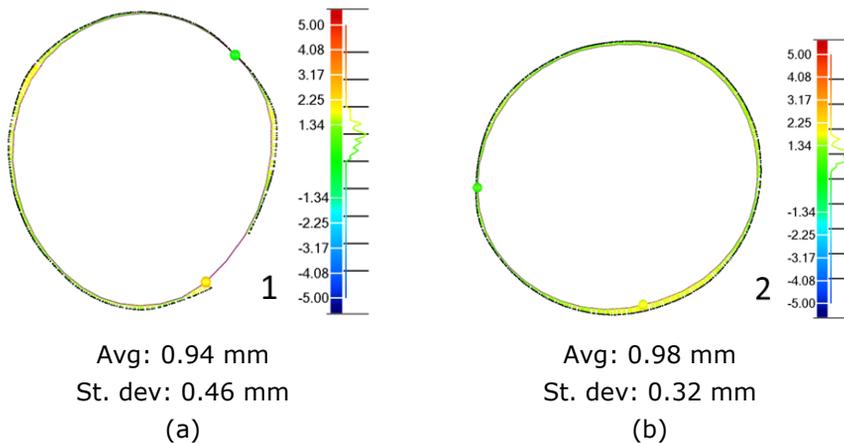
4	1.20 (1.78/-2.22)	2.94	
5	1.90 (2.55/-1.17)	3.33	

Table 1: Average and standard deviation values with histogram distribution for the 3D deviation between the scan acquisition and the relative rigged model. Five cases include one healthy subject (1), three sports injury patients (2-4), and a stroke patient (5).

For each pair model/3D scan, the 2D deviation analysis on four sections defined by notable planes was performed. Considering Figure 8(d), the planes cut the model on the arm (1), approximately in the middle of the lower arm (2), on the wrist (3), and on the palm (4). The analysis has been computed setting the model as reference, while the 3D scan as tester element. For each section, the following 2D deviations are computed: the average and the standard deviation values. Figures 8(a-d) report an example of this evaluation. Table 2 reports the results of the five persons evaluated.

Person	Arm		Lower arm		Wrist		Palm	
	Avg [mm]	St. Dev [mm]	Avg [mm]	St. Dev [mm]	Avg [mm]	St. Dev [mm]	Avg [mm]	St. Dev [mm]
1	0.81	0.65	0.78	0.51	0.55	0.53	0.11	1.50
2	0.49	0.59	0.82	0.47	0.69	0.37	0.60	0.84
3	0.94	0.46	0.98	0.32	0.41	0.25	1.39	1.11
4	1.14	0.55	1.22	0.37	0.54	0.23	1.18	2.73
5	1.35	1.04	0.36	1.50	0.71	0.65	0.26	2.57
Average	0.95	0.66	0.83	0.63	0.58	0.41	0.71	1.75

Table 2: Average and standard deviation values of 2D deviations in four different sections: arm, lower arm, wrist, and palm.



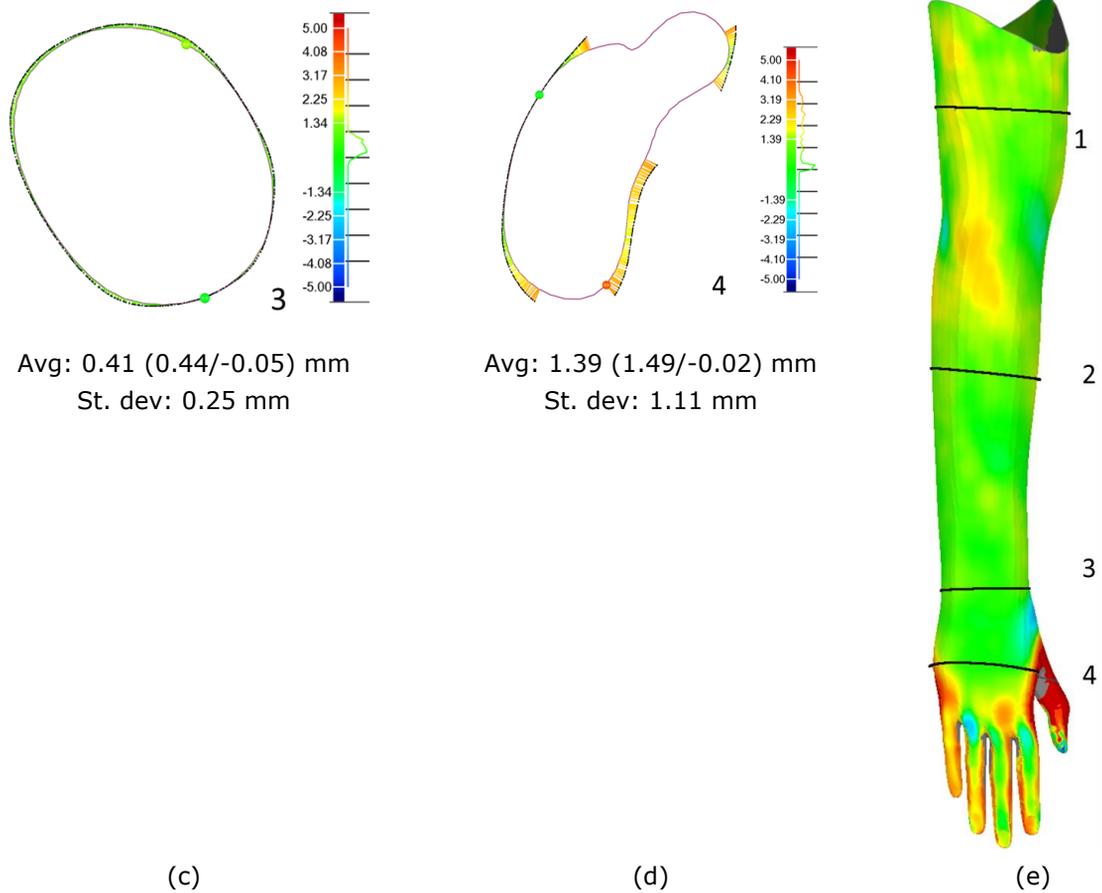


Figure 8. Example of 2D deviation analysis on the four notable planes: arm (a), the middle of the lower arm (b), wrist (c), and palm (d). The four sections are highlighted on the complete model (e). The green points on the sections represent the vertices with the minimum deviation while the red or yellow points are the vertices with the highest variation. The red line represents the rigged model surface, while the black points are the nearest vertices of the 3D scan.

6 DISCUSSION

The results highlight that the rigged model is capable of approximating the 3D acquisition of an upper limb with a limited 3D deviation. The average global variation is always lower than 2 mm. The highest values are concentrated on the hand region or in the proximity of 3D scan artifacts. It is reasonable considering that the Shrinkwrap modifier identifies the nearest vertices and tries to approximate the scan with a continuous and smooth model, but if the 3D scan presents some issues (e.g., spikes, holes, self-intersecting faces) the deviations are higher. Regarding the hand, this area is usually the one with the highest number of artifacts. Furthermore, the decision of painting skin weights to limit the influence of the modifier on the hand was done to obtain a model with fingers of the right shape and not distorted by the 3D scan errors.

To better examine the results, the 2D analysis of four significant sections has been performed. As can be seen from Table 2, the average and standard deviation values are considerably lower in the arm region. As might be expected from the previous considerations, the palm section presents the highest values ($0.71 \pm 1.75 \text{ mm}$). The arm presents an average value of 0.95 mm, but a considerably lower standard deviation ($\pm 0.66 \text{ mm}$). Also, the results of this section can be influenced

by the presence of small holes or artifacts on the mesh, as can be seen in Figure 8(a). The results continue improving with the lower arm section ($0.83\pm 0.63\text{mm}$) till the wrist ($0.58\pm 0.41\text{mm}$).

7 CONCLUSIONS

From a perspective of a digital procedure to design a rehabilitative and bespoke device, it is important to have a manifold model that represents the surface of the upper limb of a patient. The 3D scans can present holes, spikes, or intersecting surfaces caused by involuntary movements or scanning inaccuracies. This work illustrates an algorithm to generate a manifold 3D rigged model starting from the 3D scan acquisition of an upper limb of a user. The process has been assessed with five cases: one healthy subject, three patients who had sports injuries, and one stroke patient. Sport injuries and stroke patients were also chosen because they can have higher difficulties in maintaining the pose for several minutes and the scanning of their anatomies can be even more challenging. To evaluate the results, the analysis of 3D and 2D deviations between the two models has been performed. The model fits the 3D scans presenting very limited variations. Considering the properties of continuity and the smoothness of the model, the highest differences are concentrated in the areas where the number of scan issues is higher (the hand or the extremities of the arm).

This work could also be the starting point for further improvements. In particular, it would be interesting to improve the neural network by including also elbow and shoulder points in the automatic detection. Furthermore, the rigging process is usually labor-intensive, and the illustrated method to generate a bespoke 3D deformable model allows for simpler use. Thus, it can be used for the design of the controllers, but also as a virtual avatar instead of generic meshes in rehabilitative serious games.

Beatrice Aruanno, <https://orcid.org/0000-0002-8643-5915>

Sandro Barone, <https://orcid.org/0000-0003-4055-4799>

Paolo Neri, <https://orcid.org/0000-0003-0730-0893>

Alessandro Paoli, <https://orcid.org/0000-0002-1918-3033>

Francesco Tamburrino, <https://orcid.org/0000-0001-9106-0663>

ACKNOWLEDGEMENTS

The authors wish to acknowledge that this research has been supported by the European Commission as part of the H2020 program, under the grant agreement n. 856998 (PRIME-VR2 project).

REFERENCES

- [1] Aruanno, B.; Caruso, G.; Rossini, M.; Molteni, F.; Espinoza, M. C. E.; Covarrubias, M.: Virtual and Augmented Reality Platform for Cognitive Tele-Rehabilitation Based System, Computers Helping People with Special Needs. ICCHP 2020. Lecture Notes in Computer Science, vol 12376. Springer, Cham. https://doi.org/10.1007/978-3-030-58796-3_17
- [2] Aruanno, B.; Covarrubias, M.: Handy: Novel hand exoskeleton for personalized rehabilitation, Computer-Aided Design and Applications, 19(3), 2022, 405–425. <http://doi.org/10.14733/CADAPS.2022.405-425>.
- [3] Blender modifiers. <https://docs.blender.org/manual/en/latest/modeling/modifiers/index.html>
- [4] Borosán, P.; Jin, M.; Decarlo, D.; Gingold, Y.; Nealen, A. RigMesh: Automatic Rigging for Part-Based Shape Modeling and Deformation, ACM Trans Graph, 31(6), 2012, 1–9. <http://doi.org/10.1145/2366145.2366217>.
- [5] Bützer, T.; Lambercy, O.; Arata, J.; Gassert, R.: Fully Wearable Actuated Soft Exoskeleton for Grasping Assistance in Everyday Activities, Soft Robot, 8(2), 2021, 128–143. <http://doi.org/10.1089/soro.2019.0135>.

- [6] Dionne, O.; De Lasa, M.: Geodesic Voxel Binding for Production Character Meshes Animation, in Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2013, 173–180. <http://doi.org/10.1145/2485895.2485919>.
- [7] Feng, A.; Casas, D.; Shapiro, A.: Avatar reshaping and automatic rigging using a deformable model, in Proceedings of the 8th ACM SIGGRAPH Conference on Motion in Games, MIG 2015, 2015, 57–64. <http://doi.org/10.1145/2822013.2822017>.
- [8] Hachette, O.; Canezin, F.; Vaillant, R.; Mellado, N.; Barthe, L.: Automatic shape adjustment at joints for the implicit skinning, Computers and Graphics (Pergamon), 102, 2022, 300–308. <http://doi.org/10.1016/j.cag.2021.10.018>.
- [9] Intel RealSense Depth Camera D415. <https://www.intelrealsense.com/depth-camera-d415/>
- [10] Lewis, J. P.; Cordner, M.; Fong Centropolis, N.: Pose Space Deformation: A Unified Approach to Shape Interpolation and Skeleton-Driven Deformation, in SIGGRAPH '00, 2000, 165–172. <http://doi.org/10.1145/344779.344862>.
- [11] Magnenat -Thalmann, N.; Laperriere, R.; Thalmann, D.: Joint-dependent local deformations for hand animation and object grasping, in Proceedings on Graphics Interface '88, 1989, 26–33.
- [12] Makehuman - Open Source tool for making 3D characters. <http://www.makehumancommunity.org/>
- [13] MediaPipe Hands. <https://google.github.io/mediapipe/solutions/hands.html>
- [14] Neri, P.; Aruanno, B.; Barone, S.; Paoli, A.; Razionale, A. V.: Semi-automatic Point Clouds Registration for Upper Limb Anatomy, in Advances on Mechanics, Design Engineering and Manufacturing IV, Springer International Publishing, 2023, 733–742. http://doi.org/10.1007/978-3-031-15928-2_64.
- [15] Siena, F. L.; Byrom, B.; Watts, P.; Breedon, P.: Utilising the Intel RealSense Camera for Measuring Health Outcomes in Clinical Research, Journal of Medical Systems, 42(3), Springer New York LLC, 2018. <http://doi.org/10.1007/s10916-018-0905-x>.
- [16] Zhu, H. et al.: A review of point set registration: From pairwise registration to groupwise registration, Sensors (Switzerland), 19(5). 2019. <http://doi.org/10.3390/s19051191>.