

Web-based Augmented Reality with Auto-Scaling and Real-Time Head Tracking towards Markerless Neurointerventional Preoperative Planning and Training of Head-mounted Robotic Needle Insertion

Hon Lung Ho^{1*}, Yupeng Wang^{1*}, An Wang^{1*}, Long Bai¹, Hongliang Ren^{1✉}, *Senior Member, IEEE*

Abstract—Neurosurgery requires exceptional precision and comprehensive preoperative planning to ensure optimal patient outcomes. Despite technological advancements, there remains a need for intuitive, accessible tools to enhance surgical preparation and medical education in this field. Traditional methods often lack the immersive experience necessary for surgeons to visualize complex procedures and critical neurovascular structures, while existing advanced solutions may be cost-prohibitive or require specialized hardware. This research presents a novel markerless web-based augmented reality (AR) application designed to address these challenges in neurointerventional preoperative planning and education. Utilizing MediaPipe for precise facial localization and segmentation, and React Three Fiber for immersive 3D visualization, the application offers an intuitive platform for complex preoperative procedures. A virtual 2-RPS parallel positioner or “Skull-Bot” model is projected onto the user’s face in real-time, simulating surgical tool control with high precision. Key features include the ability to import and auto-scale head anatomy to the user’s dimensions and real-time auto-tracking of head movements once aligned. The web-based nature enables simultaneous access by multiple users, facilitating collaboration during surgeries and allowing medical students to observe live procedures. A pilot study involving three participants evaluated the application’s auto-scaling and auto-tracking capabilities through various head rotation exercises. This research contributes to the field by offering a cost-effective, accessible, and collaborative tool for improving neurosurgical planning and education, potentially leading to better surgical outcomes and more comprehensive training for medical professionals. The source code of our application is publicly available at https://github.com/Hill11111111ton/skullbot_web_ar.

I. INTRODUCTION

Minimally invasive neurosurgery (MIN) procedures encompass sophisticated techniques that entail precise manipulation of surgical tools within the delicate brain tissue through minimal cranial incisions [1]. Any slight misplacement of these tools during the procedure can lead to severe bleeding and significant neurological complications,

This work was supported in part by the Hong Kong Research Grants Council (RGC)-General Research Fund (GRF) 14211420, RGC - Collaborative Research Fund under grant CRF C4026-21GF, in part by the Hong Kong Research Grants Council (RGC) Research Impact Fund under grant R4020-22, and in part by the Guangdong Basic and Applied Basic Research Foundation (GBABF) under Grant 2021B1515120035. (*Corresponding author: Hongliang Ren, hren@ieee.org.*)

* H. Ho, Y. Wang, and A. Wang are the co-first authors.

¹ H. Ho, Y. Wang, A. Wang, L. Bai, and H. Ren are with the Department of Electronics Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China; and CUHK Shenzhen Research Institute, Shenzhen, China.

endangering the patient’s health [2]. Consequently, achieving meticulous localization of neural structures is crucial, typically based on computed tomography (CT) or magnetic resonance imaging (MRI) for guidance to specific target sites [1]. However, the direct correlation between CT/MRI imaging and the unique anatomy of the patient’s brain can be non-intuitive [3], necessitating the implementation of a sophisticated support system [4] [5] for precise localization and registration during surgery.

Currently, most preoperative planning and medical education still rely on static 2D images and textual descriptions, lacking the sense of immersion and interactivity [5], [6]. By providing intuitive visualization, Augmented Reality (AR) can highlight critical anatomical structures or pathologies that are hidden or challenging to find and finally guide doctors to perform surgeries more accurately [7]. AR, which can directly overlay the brain anatomy over the patient’s head, holds the potential to revolutionize the pre-surgical planning and educational frameworks within neurosurgery [8], [9]. While traditional methods have limitations in terms of time and location, AR technology in healthcare can improve safety, efficiency, doctor-patient relationships, and the invention of new surgical methods, benefiting patient outcomes and fostering innovation [10]. Therefore, the AR visualization can play a significant role in the preoperative planning of the neurosurgery.

There exist two primary types of AR applications: one relies on head-mounted devices (HMD) for observation, while the other permits direct viewing on mobile apps or websites. While hardware-based AR implementation is recognized for its costliness and lack of flexibility, web-based AR negates additional hardware requirements, facilitating effortless access to AR content on personal smartphones or tablets [11]. Moreover, when it comes to surgery, multiple users can access AR content through a single web URL, creating a collaborative platform that enhances the cooperation among surgeons [12]. Furthermore, certain medical students can actively engage in surgeries without disrupting the operating surgeon [13], thereby significantly improving medical learning opportunities. Therefore, we focus on developing and deploying a web AR application for preoperative planning and education of neurosurgery in this paper.

The main objective of our AR application is to accurately superimpose virtual models onto users’ heads. Leveraging Google’s MediaPipe [10], a robust and widely-used framework for face landmark detection and segmentation, develop-

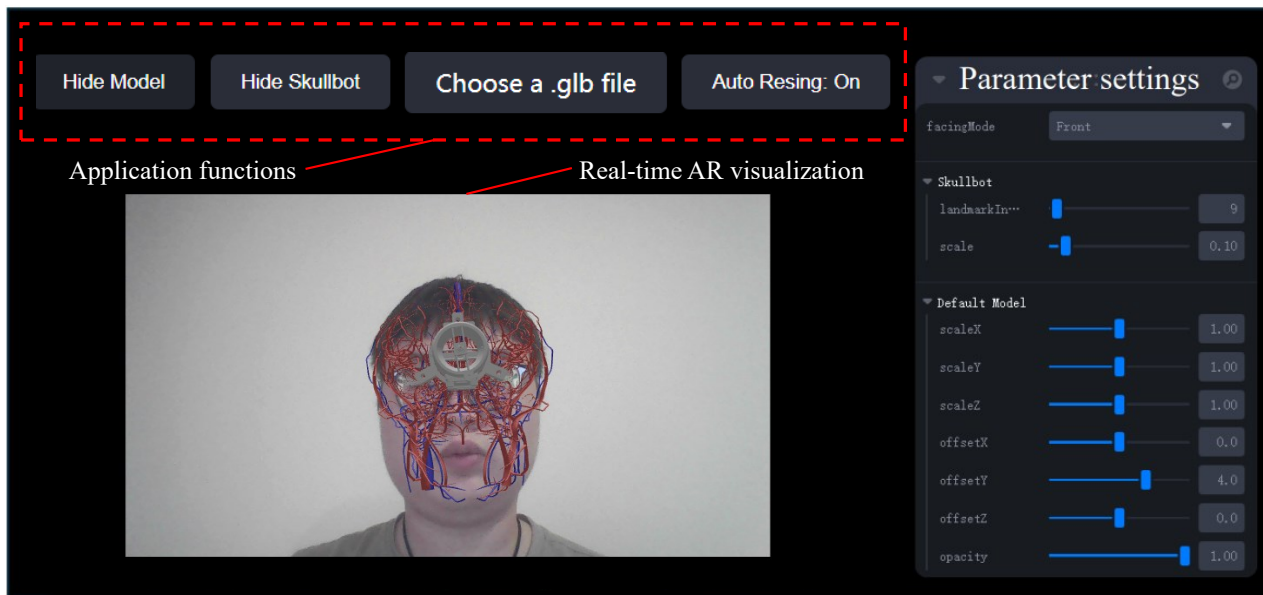


Fig. 1: Layout of the web-based AR application. The interface includes three distinct regions, each serving a specific purpose. The upper region is the function block with several buttons, which controls the import/hide of the 3D model and enables/disables auto-scaling. The middle region is a window for displaying the AR content. The right region is a panel for parameter settings, which can be used to manually modify the imported model.

ers can efficiently manage resources while maintaining high-quality results [14]. Renowned for its precision and utility in medical applications, MediaPipe has been extensively validated through research, demonstrating 95% consistency limits and comparable accuracy to traditional measurement tools such as the universal goniometer and digital angle ruler [15]. Integrating artificial intelligence (AI) in AR has further streamlined automated tracking, significantly reducing preparation time for surgical procedures [16].

In minimally invasive neurosurgery, two critical factors are essential: precise visualization of target brain structures and accurate control over surgical instruments. Integrating robotics in surgery has significantly enhanced operating efficiency and safety by reducing operating times, minimizing blood loss, and accelerating patient recovery [17] [18]. The 2-RPS parallel positioner (Skull-Bot) is a notable head-mounted robot for needle insertion, achieving remarkable open-loop positioning accuracy within ± 1 mm in neurosurgical procedures [15] [19]. By combining the Skull-Bot with web-based augmented reality visualization, we aim to further refine the preoperative process with improved intuitiveness.

As shown in Fig. 1, this paper introduces a novel marker-free web-based AR application that can assist neurosurgical preoperative planning and training such as head-mounted robotic needle insertion procedures. By leveraging facial feature detection and immersive 3D visualization, our system enables users to import and overlay 3D head anatomy structures in real-time, simulating precise surgical tool placement. This innovative approach aims to streamline pre-surgical training and planning, ultimately reducing the time required for surgeons and medical students to become proficient

with specific tools. The contributions of this research are threefold:

- Development of a markerless web-based AR platform: Our system integrates facial feature detection and achieves immersive 3D visualization, providing an intuitive interface for neurosurgical preoperative planning and education.
- Auto-scaling and tracking capabilities: The application features auto-rescaling to accommodate different head dimensions and enables fast and precise auto-tracking of head movements, enhancing the overall user experience.
- Evaluative validation: A qualitative and quantitative assessment involving three users demonstrates the high performance and quality of our system with over 80% overlap IoU, showcasing its potential to improve neurosurgery procedures.

II. METHODOLOGY

A. Application Overview

The application was developed using Three.js [20], a JavaScript library that enables the creation and rendering of 3D objects in web browsers via WebGL. This powerful tool simplifies the development of complex visual effects, allowing for the seamless integration of 3D elements into web-based interfaces. With Three.js, developers can easily create interactive 3D experiences by scaling, rotating, and animating 3D objects directly within the browser. Furthermore, React Three Fiber¹ (R3F) was employed as a React renderer for Three.js [20]. R3F provides an intuitive interface

¹<https://github.com/pmndrs/react-three-fiber>

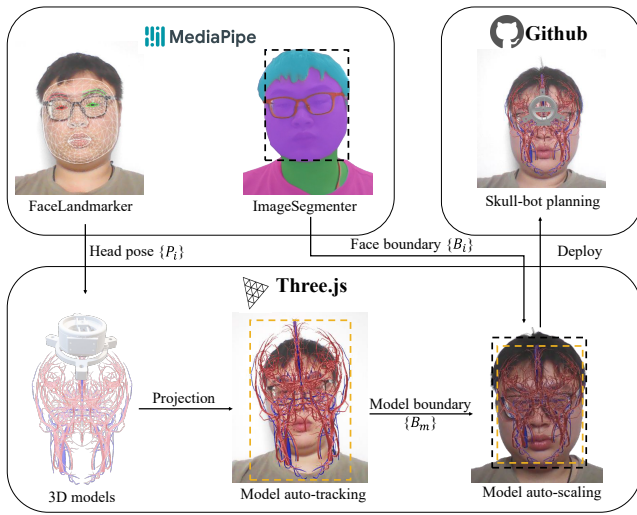


Fig. 2: Overall working mechanism of our AR application.

to work with Three.js, seamlessly integrating it into the React ecosystem. This integration not only streamlines the development process but also leverages React’s state management capabilities, significantly reducing development time.

The development environment was established by integrating two key libraries: React Three Fiber (R3F) for 3D scene management and MediaPipe [10] for facial landmark detection. A webcam feed was positioned on a distant plane using R3F, while a 3D model was placed closely in front of the perspective camera offered by Three.js. This setup effectively simulates an augmented reality (AR) environment, rendering the 3D model between the viewer and the webcam plane. The resulting AR simulation enables users to interact with virtual objects as if they were real-world objects, enhancing the overall user experience. The developed application can be deployed locally or hosted by a server. We have deployed it on GitHub² to provide users with convenient access to the AR application from anywhere with an internet connection and a webcam. This deployment strategy not only facilitates user engagement but also promotes accessibility, allowing users to explore the AR environment at their own pace.

The AR application consists of three main components, as illustrated in Fig. 1. Firstly, an intuitive interface is provided through a set of buttons that enable users to access various functions within the application. This includes the ability to import different 3D head models, which can be customized to display various anatomy structures for educational purposes. Additionally, the skull-bot model can be visualized, facilitating planning and preparation for surgical operations. The imported 3D model can also be adjusted using the parameters panel, built utilizing Leva³, a powerful React widget library that streamlines parameterization and configuration. This feature allows users to fine-tune the model’s appearance and behavior, ensuring a precise representation of the anatomy.

²https://hilllllllllton.github.io/fyp_AR/

³<https://github.com/pmdrs/leva>

Finally, the camera view is seamlessly overlaid with the 3D model in real-time, providing an immersive augmented reality experience.

B. AR Application Implementation

Accurate pre-operative planning requires a precise overlay of the virtual 3D model onto the user’s head. This necessitates the implementation of a robust auto-tracking system, which continuously monitors the user’s head position to ensure seamless alignment with the model. Moreover, the size of the imported 3D virtual model may not match the user’s head dimensions, requiring an adaptive solution that adjusts the model’s scale accordingly. To address this challenge, an auto-scaling mechanism was developed, which scales the model in real-time based on the user’s head size. The successful implementation of these two core functions is critical for a seamless user experience. The general workflow is visualized in Fig. 2, and we will elaborate on it in detail in the following sections.

1) *Head auto-tracking*: The FaceLandmarker model from MediaPipe processes webcam footage to detect facial landmarks in real-time. These landmarks serve as a guide for positioning and orienting the 3D model within the AR environment, enabling precise registration between the virtual model and the user’s face. To ensure seamless tracking of the face, we employ the face mesh, a 4×4 matrix P_i , to dynamically adjust the 3D model’s orientation and position in real-time. This sophisticated approach ensures that the model accurately overlays and moves with the user’s face, providing an immersive AR experience.

Based on the pose P_i , as well as the imported mesh, the surface points P_w can be determined. The application projects the surface points onto a 2D plane, allowing users to visualize it within the web interface. To achieve this projection, we assume that the 2D plane is located at z_c ahead of the camera. By leveraging the properties of perspective projection, we can derive the following equation:

$$z_c p_c = K T P_w, \quad (1)$$

where p_c represents the 2D location of the point in pixels, K is the camera intrinsics matrix that encodes information about the camera’s focal length and image sensor dimensions, T is the transformation from the camera frame to the world frame, and $P_w = [X, Y, Z]^T$ denotes the 3D position of the model in the world coordinate system.

Given that we assume the camera frame is equivalent to the world frame, the transformation matrix T becomes an identity matrix, and the depth value z_c reduces to the 3D coordinate Z . Furthermore, we utilize the default camera intrinsics provided by three.js. Under these conditions, the equation simplifies to:

$$p_c = K P_w / z_c = K P_n, \quad (2)$$

where the normalized 3D position P_n can be defined as:

$$P_n = \begin{bmatrix} X/Z \\ Y/Z \\ 1 \end{bmatrix}. \quad (3)$$

This normalized form can be used to complete the projection and achieve accurate 2D rendering.

2) *Model auto-rescaling*: When the 3D model is projected onto the 2D image, it is bounded by a rectangular B_m . However, its width (w_m) and height (h_m) may not accurately fit the user’s head. To address this issue, we can re-scale the virtual model to achieve a better match between the virtual and real worlds.

The ImageSegmenter from MediaPipe is used to segment the user’s head in real-time, providing another boundary B_i . The width (w_i) and height (h_i) of this segmentation are more accurately tailored to fit the user’s head. To scale the 3D model boundaries to match these dimensions, we calculate the scale factors:

$$s_w = w_i/w_m, \quad (4)$$

$$s_h = h_i/h_m. \quad (5)$$

With these scale factors, the projection equation becomes:

$$p_c = KSP_n, \quad (6)$$

where S is a diagonal scale matrix defined as:

$$S = \begin{bmatrix} s_w & 0 & 0 \\ 0 & s_h & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

This allows us to auto-scale the 3D model based on head size, achieving a precise virtual-to-real matching for different users.

3) *Skull-Bot visualization*: In addition to the previously mentioned features, our system also utilizes the skull and the 2-RPS parallel positioner (Skull-Bot) [15]. The Skull-Bot is particularly well-suited for applications requiring exact placement or manipulation within constrained spaces, making it an ideal choice for medical or surgical procedures where precision is crucial. Given its precise positioning requirements, setting up the Skull-Bot can be a complex task. However, our system allows users to simulate the placement of the Skull-Bot for pre-planning surgery and educational training purposes before actually using the machine. This capability enables users to practice and refine their skills in a virtual environment, reducing the risk of errors and improving overall efficiency.

C. Additional Functions

In addition to its core capabilities, our system includes several additional features designed to enhance the user experience.

1) *Model visibility toggle*: To further empower users, we have implemented a feature that allows them to toggle the visibility of 3D models on demand. By incorporating a user-friendly button into the interface, users can easily hide or show models as needed, making the application adaptable to different learning and exploration scenarios. This simple yet effective functionality provides users with greater control over their interaction with the application, enabling them to focus on specific aspects of an augmented reality (AR) environment or simplify complex scenes for clearer understanding and guidance.

2) *Camera selection capabilities*: This application provides a flexible camera selection feature that allows users to seamlessly switch between the front and rear cameras on their device, whether it is a tablet or mobile phone. This capability is particularly valuable in educational and presentation settings, where users often need to toggle perspectives quickly.

3) *Manual model customization*: In addition to auto-rescaling, our application offers a high degree of manual control over the 3D model, allowing users to personalize their experience and achieve precise alignment. With our intuitive tools, users can manually scale, position, and orient the model in three-dimensional space, ensuring that it perfectly matches their face. Furthermore, users can adjust the model’s transparency to see through it to the underlying physical structure. This feature is particularly useful in educational environments or real-world surgery, where visualizing underlying anatomy is crucial.

4) *Custom model import*: To expand its versatility and cater to diverse educational and practical applications, our software allows users to import their own GLB (.glb) models. This feature empowers users to create highly customizable AR experiences that can accommodate specialized or proprietary models. With this capability, users can seamlessly integrate their own models into the existing interface, giving them access to the same level of control as preloaded models. Specifically, they can position, scale, and adjust the transparency of their imported model with ease. This feature is particularly beneficial for professionals in the medical field, who can create personalized simulations and educational models tailored to their specific needs.

III. EXPERIMENTS AND RESULTS

A. Experiment Setup

The AR application was deployed on a computer with an NVIDIA RTX3050 GPU, utilizing 0.4GB of shared memory resources. A detailed 3D model of blood vessels in the head was imported into the application as the test subject. To evaluate its performance in real-world scenarios, three participants with varying head sizes were recruited for the experiment. The initial setup involved overlapping the virtual model on the participant’s head and scaling it to match their individual head size. Subsequently, each participant rotated their head around its axis in pitch, yaw, and roll direction by roughly 45 degrees, as demonstrated in the left of Fig. 4. Finally, to assess the application’s tracking error, we drew the boundaries of both the model (B_m) and the participant’s head (B_i).

B. Evaluation Metrics

The experiment employed two key metrics to evaluate the AR application’s performance: positional accuracy and Intersection over Union (IOU). The former is calculated as the difference between the width and height of the participant’s head and the projected 3D model. IOU measures the overlap performance by comparing the intersection area of the model boundary (B_m) with the participant’s head boundary (B_i).

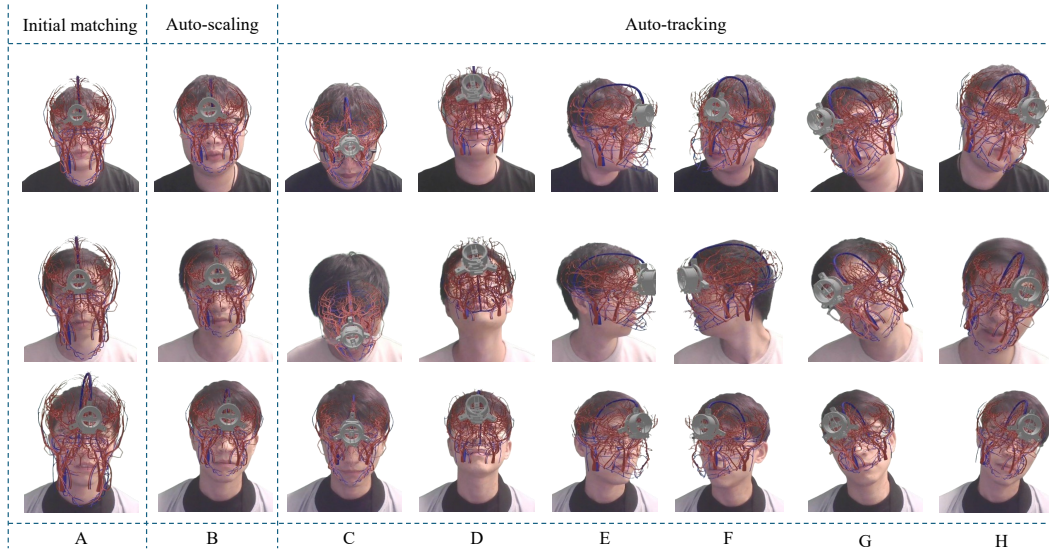


Fig. 3: Visualization of the scaling and tracking results. The virtual model can auto-scale itself (A to B) and track head movements automatically. The participants rotated their heads in the direction of Pitch (C, D), Yaw (E, F), and Roll (G, H).

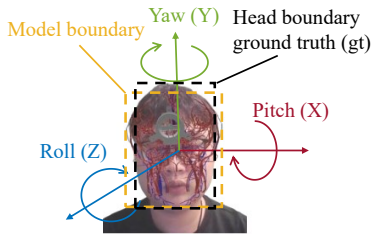


TABLE I. The evaluation of the overlapping performance

Performance	Pitch (%)	Yaw (%)	Roll (%)	Overall (%)
Width error	10.01	11.60	11.32	10.09 ± 5.53
Height error	8.69	7.35	7.19	7.60 ± 5.53
IOU	80.07	80.23	80.87	80.87 ± 4.42

Fig. 4: Experimental quantification results. The user's head is rotated in 3 DOF to roughly 45 degrees. For each DOF, the overlap error of width, height, and IOU are calculated. The application generally achieves decent tracking performance.

A higher IOU value indicates better overlapping accuracy. In this experiment, we used the participant's head boundary (B_i) as the ground truth (gt) for evaluation purposes.

1) *Width Error And Height Error*: To assess the accuracy of the model scaling, we evaluated its performance in two dimensions: width and height. The scale factor was applied to match the initial model size with that of the user's head. We defined the error in width (E_w) and height (E_h) direction as follows:

$$E_w = w_m/w_i \times 100\%, \quad (7)$$

$$E_h = h_m/h_i \times 100\%, \quad (8)$$

where w_i and h_i are the width and height of the user's head boundary (B_i), respectively, and w_m and h_m are the width and height of the model boundary (B_m), respectively.

2) *Intersection over Union*: In addition to evaluating the scaling accuracy, we also calculated the IOU metric to assess the application's tracking ability. The IOU is defined as:

$$IOU = \frac{B_i \cap B_m}{B_i \cup B_m} \quad (9)$$

The IOU value represents the ratio of the intersection area to the union area of the two boundaries.

C. Results And Analysis

The experiment revealed promising results, as demonstrated in Fig. 3. Upon startup, the application would first make an initial match between the head and the virtual model. The virtual model was then scaled to the correct size based on the segmentation result, as demonstrated from columns A to B in Fig. 3. The scaled model effectively tracked the head rotations, achieving satisfactory outcomes despite variations in face shape, hairstyle, and head size among participants.

The analysis of error metrics revealed moderate disparities between the scaled model and actual head measurements. Specifically, as shown in Table I, the width error was reported as $10.09 \pm 5.53\%$ (mean ± standard deviation), while the height error was $7.60 \pm 5.53\%$. Rotations in the yaw direction had the most significant effect on width error, suggesting that this axis is particularly challenging for accurate model scaling. Conversely, rotations in the pitch direction exerted a more pronounced influence on height error.

Moreover, the IOU values as shown in Table I revealed a relatively high level of agreement between the ground truth and model boundary across different dimensions. The

average IOU value of $80.87 \pm 4.42\%$ suggests that the application effectively overlays the virtual model onto the user's head. Notably, rotation in various directions has a similar impact on the IOU metric, indicating robustness against different head orientations.

IV. DISCUSSION AND CONCLUSION

The development and deployment of our AR application for neurointerventional preoperative planning mark a significant step forward in the integration of medical technology and education. By harnessing the power of MediaPipe [10] for facial landmark detection and segmentation, our immersive marker-free 3D visualization approach offers a novel and engaging platform for training and preparation in neurosurgery procedures, i.e., head-mounted robotic needle insertion with Skull-Bot. The current emphasis of this paper is on implementing the registration of a virtual head model in web-based AR. In the future, the real Skull-Bot can be positioned according to AR planning, showcasing its feasibility in actual surgical scenarios.

The experiments conducted on diverse participants demonstrate that the application performs reasonably well in auto-scale accuracy, with IOU scores indicating a strong degree of overlap between the ground truth and model boundary. While these results are promising, further refinements to the scaling algorithm, optimization of the boundary detection method, and fine-tuning of localization accuracy are crucial for continued improvement.

Notably, our application's reliance on facial tracking provided by MediaPipe [10] poses limitations in certain situations, such as hind-brain surgery training where the face may not be visible or accessible. To broaden its utility and versatility across various surgical procedures, future developments should focus on reducing this dependency and exploring alternative tracking methods or technologies.

In conclusion, the successful integration of auto-scale and auto-tracking capabilities into our markerless web-based AR application has significant implications for preoperative planning, enhancing the situation awareness of surgeons and medical professionals. By addressing the aforementioned limitations and building upon existing successes, we envision a more comprehensive and accessible platform for neurosurgery preoperative planning, ultimately enhancing patient care and outcomes.

ACKNOWLEDGMENT

We would like to express our gratitude to Dr. Lalithkumar Seenivasan for his invaluable discussions and to Ruijie Tang for participating in the experiments.

REFERENCES

- [1] H. Su, W. Shang, G. Li, N. Patel, and G. S. Fischer, "An mri-guided telesurgery system using a fabry-perot interferometry force sensor and a pneumatic haptic device," *Annals of biomedical engineering*, vol. 45, pp. 1917–1928, 2017.
- [2] A. De Benedictis, A. Trezza, A. Carai, E. Genovese, E. Procaccini, R. Messina, F. Randi, S. Cossu, G. Esposito, P. Palma *et al.*, "Robot-assisted procedures in pediatric neurosurgery," *Neurosurgical focus*, vol. 42, no. 5, p. E7, 2017.
- [3] B. Baheti, D. Waldmannstetter, S. Chakrabarty, H. Akbari, M. Bilello, B. Wiestler, J. Schwarting, E. Calabrese, J. Rudie, S. Abidi *et al.*, "The brain tumor sequence registration challenge: establishing correspondence between pre-operative and follow-up mri scans of diffuse glioma patients," *arXiv preprint arXiv:2112.06979*, 2021.
- [4] L. Joskowicz, R. Shamir, M. Freiman, M. Shoham, E. Zehavi, F. Umansky, and Y. Shoshan, "Image-guided system with miniature robot for precise positioning and targeting in keyhole neurosurgery," *Computer Aided Surgery*, vol. 11, no. 4, pp. 181–193, 2006.
- [5] J. Shapey, T. Dowrick, R. Delaunay, E. C. Mackle, S. Thompson, M. Janatka, R. Guichard, A. Georgoulas, D. Pérez-Suárez, R. Bradford *et al.*, "Integrated multi-modality image-guided navigation for neurosurgery: open-source software platform using state-of-the-art clinical hardware," *International journal of computer assisted radiology and surgery*, vol. 16, pp. 1347–1356, 2021.
- [6] J. Guo, Q. Guo, M. Feng, S. Liu, W. Li, Y. Chen, and J. Zou, "The use of 3d video in medical education: A scoping review," *International Journal of Nursing Sciences*, 2023.
- [7] L. Qian, J. Y. Wu, S. P. DiMaio, N. Navab, and P. Kazanzides, "A review of augmented reality in robotic-assisted surgery," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 1, pp. 1–16, 2019.
- [8] A. H. Sadeghi, S. El Mathari, D. Abjigitova, A. P. Maat, Y. J. Taverne, A. J. Bogers, and E. A. Mahtab, "Current and future applications of virtual, augmented, and mixed reality in cardiothoracic surgery," *The Annals of thoracic surgery*, vol. 113, no. 2, pp. 681–691, 2022.
- [9] K. Lee, "Augmented reality in education and training," *TechTrends*, vol. 56, pp. 13–21, 2012.
- [10] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee *et al.*, "Mediapipe: A framework for perceiving and processing reality," in *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, vol. 2019, 2019.
- [11] X. Qiao, P. Ren, S. Dustdar, L. Liu, H. Ma, and J. Chen, "Web ar: A promising future for mobile augmented reality—state of the art, challenges, and insights," *Proceedings of the IEEE*, vol. 107, no. 4, pp. 651–666, 2019.
- [12] I. A. Al Hafidz, S. Sukaridhoto, M. U. H. Al Rasyid, R. P. N. Budiarti, R. R. Mardhotillah, R. Amalia, E. D. Fajrianti, and N. A. Satrio, "Design of collaborative webxr for medical learning platform," in *2021 International Electronics Symposium (IES)*. IEEE, 2021, pp. 499–504.
- [13] M. Javaid and A. Haleem, "Virtual reality applications toward medical field," *Clinical Epidemiology and Global Health*, vol. 8, no. 2, pp. 600–605, 2020.
- [14] A. Latreche, R. Kelaiaia, A. Chemori, and A. Kerboua, "Reliability and validity analysis of mediapipe-based measurement system for some human rehabilitation motions," *Measurement*, vol. 214, p. 112826, 2023.
- [15] X. Xiao, H. Gao, C. Li, L. Qiu, K. S. Kumar, C. J. Cai, B. S. Bhola, N. K. K. King, and H. Ren, "Portable body-attached positioning mechanism toward robotic needle intervention," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 2, pp. 1105–1116, 2020.
- [16] C. K. Sahu, C. Young, and R. Rai, "Artificial intelligence (ai) in augmented reality (ar)-assisted manufacturing applications: a review," *International Journal of Production Research*, vol. 59, no. 16, pp. 4903–4959, 2021.
- [17] B. Yang, S. Roys, U.-X. Tan, M. Philip, H. Richard, R. P. Gullapalli, and J. P. Desai, "Design, development, and evaluation of a master-slave surgical system for breast biopsy under continuous mri," *The International journal of robotics research*, vol. 33, no. 4, pp. 616–630, 2014.
- [18] Z. Chen, K. Fan, L. Cruciani, M. Fontana, L. Muraglia, F. Ceci, L. Travaini, G. Ferrigno, and E. De Momi, "Toward human-out-of-the-loop endoscope navigation based on context awareness for enhanced autonomy in robotic surgery," *IEEE Transactions on Medical Robotics and Bionics*, 2024.
- [19] C. Li, N. K. K. King, and H. Ren, "A skull-mounted robot with a compact and lightweight parallel mechanism for positioning in minimally invasive neurosurgery," *Annals of biomedical engineering*, vol. 46, pp. 1465–1478, 2018.
- [20] B. Danchilla and B. Danchilla, "Three.js framework," *Beginning WebGL for HTML5*, pp. 173–203, 2012.