COMILLAS

UNIVERSIDAD PONTIFICIA

ICAI

BACHELOR'S DEGREE IN MATHEMATICAL ENGINEERING AND ARTIFICIAL INTELLIGENCE

BACHELOR'S THESIS

USING DEPTH FOR ENHANCING SEMANTIC SEGMENTATION FOR EXTENDED REALITY APPLICATIONS

Author: Sofía Pedrós Tobaruela
Director: Ester González-Sosa
Co-Director: David Contreras Bárcena

Madrid
June 2025

I declare, under my own responsibility, that the Project submitted under the title

Using depth for enhancing semantic segmentation for Extender Reality Applications

at the School of Engineering – ICAI of the Universidad Pontificia Comillas in the academic year 2024/25 is my own work, original and unpublished, and has not been submitted previously for any other purpose.

The Project is not plagiarized, either in whole or in part, and any information taken from other sources has been properly cited.

Signed: Sofía Pedrós Tobaruela Date: 11/6/2025

Authorized for submission

THE PROJECT SUPERVISORS

Signed: Ester Gonzalez-Sosa Date: 12/06/2025

Signed: David Contreras Bárcena Date: 11/6/2025

Firstly, I would like to thank my family for their unconditional support. I would also like to thank my director, Ester, and co-director David, for their guidance and invaluable help throughout the development of this project.

Using Depth for Enhancing Semantic Segmentation for Extended Reality Applications

Author: Pedrós Tobaruela, Sofía.
Supervisor: González-Sosa, Ester.
Co-Supervisor: Contreras Bárcena, David.
Collaborating Entity: Nokia Extended Reality Lab

Abstract

Extended Reality (XR) technologies offer immersive experiences blending virtual and real environments. For an optimal experience, users must feel integrated in the scene. This requires an accurate depiction of the user, often achieved via video-based self-avatars generated through egocentric semantic segmentation. We introduce a real-time depth-enhanced model, trained on a RGB-D egocentric dataset, and a reconstruction of the user's video-based self-avatar as a 3D point cloud. We show that using depth improves segmentation performance, with a 13.75% relative increase in mIoU and a 39.60% gain in MOS, and that representing the user in 3D enhances distance perception in XR.

Keywords: Semantic segmentation, transformers, CNNs, Extended Reality, Egocentric vision, RGB-D

1 Introduction

Extended Reality (XR) is a term that encompasses various immersive technologies such as Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR). In particular, MR seamlessly blends virtual and real environments [7]. A critical aspect of immersion in MR is the presence factor - the perception of "being there" within the virtual space [5]-. One key element to improve this sense of presence is to create an accurate representation of the user in the virtual world. This can be achieved through a video-based self-avatar, which involves segmenting the user's body from a first-person point of view and then rendering it in the virtual environment. Previous implementations have used Machine Learning (ML) algorithms to generate these self-avatars in real-time from an egocentric RGB image [6][2]. However, segmentation quality can often be improved, as background objects with skin-like tones tend to be classified as false positives. Moreover, rendering a 2D self-avatar into a virtual scene, does not provide an accurate perception of distances. These errors suggest that depth information may improve segmentation accuracy, reducing the rate of false positives, and enable 3D reconstruction of the user as a point cloud. This can potentially improve the quality of the MR experience thereby increasing the presence factor.

2 Project Definition

This work builds upon previous work of egocentric segmentation in MR by generating video-based self-avatars in real-time using RGB-D data and leveraging depth to create a 3D representation of the user within a MR environment.

All primary and secondary objectives were achieved, resulting in both technical and experiential contributions. The completed objectives can be summarized as follows:

- Explore a new sensor (RealSense D435) that captures monocular RGB and depth frames, including the calibration of both cameras and the physical integration in the MR headset.
- Update existing datasets to include depth frames.
- Train real-time semantic segmentation deep learning architectures with RGB data and RGB-D data (Thundernet and AsymFormer).

- Design and develop a Unity-based MR application capable of rendering the segmented user as a 3D point cloud, including the communication with a deployed segmentation server.
- A complete subjective evaluation comparing conditions with and without depth information.

These efforts have translated into four main technical contributions:

- A multimodal algorithm capable of segmenting egocentric bodies based on the AsymFormer architecture [1]. To the best of our knowledge, this is the first network able to segment egocentric bodies in real time using RGB-D input.
- Creation of an RGB-D dataset for egocentric semantic segmentation, through the extension of the Egocentric Bodies Dataset [3].
- Integration of ML algorithms in a MR environment based on Unity.
- A double evaluation -qualitative and quantitative- demonstrating how depth affects the segmentation performance and the user's perception within the environment.

3 System Description

To compare how depth enhances segmentation results, two real-time semantic segmentation models were trained: Thundernet and AsymFormer. Both models were trained with the Egocentric Bodies Dataset [3], which was extended in this project into an RGB-D version by adding the corresponding depth frames. Thundernet [8] is light-weight network based on a U-Net architecture with CNNs to extract key features. In contrast, AsymFormer [1] is a novel network that processes depth and color inputs concurrently, offering better segmentation accuracy at the cost of higher inference time.

To integrate these ML models into MR, this project implements the pipeline shown in Figure 1. The user's body and surroundings are captured with a RealSense D435 attached to the headset with a custom 3D-printed camera cover. These frames are sent to a semantic segmentation model (AsymFormer or Thundernet) via Unity. The model returns a mask of the user's body which Unity renders as a 3D point cloud or a 2D overlay in MR, depending on the desired experience.

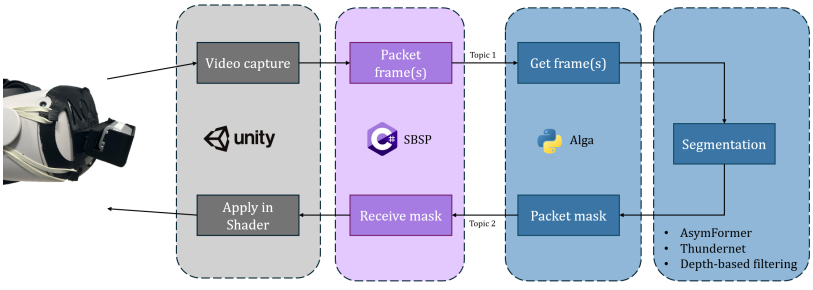


Figure 1: Complete E2E pipeline for the final application.

The effect of using depth in a MR environment was assessed objective and subjectively. Two parallel evaluations were designed. The first one is to determine whether incorporating depth information improves segmentation quality, and if the reduction of efficiency that comes with using more complex networks is justified to improve the final experience. It is a video-based experience where users will be presented several segmentation outputs and asked to rate which one they perceive as better. These ratings will be compared with mIoU scores computed over some of the video frames. The second evaluation examines if representing the user with a point cloud (using depth data) enhances the sense of presence compared

to a standard 2D representation or does not provide a significant improvement. This second experiment is embedded within a MR environment, where the users will interact with a virtual environment. Their body will be segmented using different models and two visualizations modes will be available: one that showcases the segmentation result as a point cloud and the other will plot it in 2D. Users will assess whether the increased depth perception enhances their overall experience, even if the point cloud representation may flicker or exhibit blind spots due to the depth sensor limitations. The models that will be compared are Thundernet, AsymFormer and a depth-based filter (without ML).

4 Results

Regarding segmentation accuracy, AsymFormer outperformed ThunderNet by approximately 3% in mIoU in the Egocentric Bodies dataset, highlighting the benefits of integrating depth information. This improvement is mainly attributed to depth aiding in the discard of false positives in the background and improving segmentation in color-ambiguous regions.

In the video evaluation, AsymFormer led subjectively, with a MOS of 4.23/5, and objectively, with an IoU of 0.82, using SAM-based labels [4] as ground truth.

In the Unity evaluation, the results varied. While the point cloud enhanced distance perception, the best configuration was Thundernet with a 2D visualization. According to user feedback, this was mainly due to imperfections of the point cloud, such limited FOV of the RealSense, lower definition and latency issues, particularly with AsymFormer. Nevertheless, the depth-filtering alternative rendered with a point cloud emerged as a strong candidate, as it provides depth awareness without delays and blinking false positives. Some experienced participants preferred this model, noting its capability to interact with a 3D virtual scene. All results can be summarized in Table 1.

Model	**EgoBodies dataset**		**Video evaluation**		**Unity evaluation**	
	Test IoU	Speed (s)	IoU	MOS	MOS (Overall)	MOS (Distance)
Thundernet Mono	0.895	0.0199	0.72	3.46	–	–
Thundernet Stereo	0.814	0.0199	0.68	3.03	**3.91**	4.25
Depth Filter	–	0.0000	0.32	2.52	3.55	**4.73**
AsymFormer (RGB-D)	**0.926**	0.0329	**0.82**	**4.23**	3.02	4.60

Table 1: Summary of performance across the EgoBodies dataset, video-based evaluation, and Unity-based evaluation (overall and distance perception).

5 Conclusions

This project demonstrates that depth-enhanced egocentric segmentation significantly improves both segmentation accuracy and the immersive experience in MR applications. The 3D point cloud visualization improved depth perception, especially in scenarios involving spatial interaction. However, the limitations of current depth sensors—such as missing or unstable points and lack of definition—also affect the quality of the 3D reconstruction. Additionally, the high computational cost of models like AsymFormer may not justify their use in all contexts. For non-interactive scenarios, simpler 2D segmentation may suffice. Future work could explore alternative 3D representations that are more visually stable, and alternative real-time optimizations for using more complex models. Improvements in technology, such as depth sensors with a wider FOV, could further enhance the utility of depth-aware segmentation in MR environments by providing an even more realistic self-avatar.

References

[1] Siqi Du, Weixi Wang, Renzhong Guo, Ruisheng Wang, and Shengjun Tang. Asymformer: Asymmetrical cross-modal representation learning for mobile platform real-time rgb-d semantic segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7608–7615, 2024. doi: 10.1109/CVPRW63382.2024.00756.

[2] Ester González-Sosa, Guillermo Robledo, Diego González Morín, Pablo Perez-Garcia, and Álvaro Villegas. Real time egocentric object segmentation: Thu-read labeling and benchmarking results. *ArXiv*, abs/2106.04957, 2021. URL `https://api.semanticscholar.org/CorpusID:235377314`.

[3] Ester González-Sosa, Andrija Gajic, Diego González Morín, Guillermo Robledo, Pablo Pérez, and Álvaro Villegas. Real time egocentric segmentation for video-self avatar in mixed reality. *ArXiv*, abs/2207.01296, 2022. URL `https://api.semanticscholar.org/CorpusID:250264412`.

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[5] Kwan Min Lee. Presence, explicated. *Communication Theory*, 14(1):27–50, 2004. doi: 10.1111/j.1468-2885.2004.tb00302.x. URL `https://academic.oup.com/ct/article/14/1/27/4110793`.

[6] Diego Gonzalez Morin, Ester Gonzalez-Sosa, Pablo Perez, and Alvaro Villegas. Full body video-based self-avatars for mixed reality: from e2e system to user study. *Virtual Reality*, 27(3):2129–2147, 2023. URL `https://doi.org/10.1007/s10055-023-00785-0`.

[7] Richard Skarbez, Missie Smith, and Mary C Whitton. Revisiting milgram and kishino's reality-virtuality continuum. *Frontiers in Virtual Reality*, 2:647997, 2021.

[8] Wei Xiang, Hongda Mao, and Vassilis Athitsos. Thundernet: A turbo unified network for real-time semantic segmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1789–1796, 2019. doi: 10.1109/WACV.2019.00195.

Usando Profundidad Para Mejorar la Segmentación Semántica en Aplicaciones de Realidad Extendida

Autor: Pedrós Tobaruela, Sofía.
Director: González-Sosa, Ester.
Co-Director: Contreras Bárcena, David.
Entidad colaboradora: Nokia Extended Reality Lab

Resumen

Las tecnologías de Realidad Extendida (XR) ofrecen experiencias inmersivas que mezclan entornos reales y virtuales. Para una experiencia óptima, el usuario debe sentirse integrado en la escena. Esto requiere una representación del usuario, que puede ser un avatar basado en vídeo generado por segmentación semántica. Presentamos un modelo en tiempo real, mejorado con profundidad, entrenado con un conjunto de datos egocéntrico RGB-D y una reconstrucción del avatar como una nube de puntos 3D. Demostramos que incorporar profundidad mejora la segmentación (en un 13.75% de mIoU y un 39.6% de MOS), y que representar al usuario en 3D mejora la percepción de distancias en XR.

Palabras clave: Segmentación semántica, transformers, CNNS, Realidad Extendida, Visión egocéntrica, RGB-D

1 Introducción

La Realidad Extendida (XR) se refiere a un término que engloba varias tecnologías inmersivas, como Realidad Virtual (VR), Realidad Aumentada (AR) y Realidad Mixta (MR). En concreto, la MR mezcla los entornos reales y virtuales [7]. Un aspecto crítico para la inmersión en MR es lo que se llama factor de presencia - la percepción de "estar ahí" en la escena virtual [5]-. Un elemento clave para mejorar el factor de presencia es crear una representación realista del usuario en el mundo virtual. Esto se puede conseguir con un avatar basado en vídeo, que implica segmentar el cuerpo del usuario en primera persona y después visaulizarlo en la escena virtual. Algunas implementaciones previas han usado Machine Learning (ML) para crear estos avatares en tiempo real usando imágenes RGB egocéntricas [6][2]. A pesar de esto, la segmentación se puede mejorar, porque suele haber objetos en el fondo que se suelen clasificar como falsos positivos (especialmente objetos con tonalidades parecidas a la piel). Además, pintar un avatar 2D en un mundo 3D virtual hace que las distancias no se perciban bien. Estos errores sugieren que se puede usar la profundidad tanto para mejorar la precisión de la segmentación (reduciendo los falsos positivos), como para pintar el cuerpo del usuario en 3D como una nube de puntos. Potencialmente, esto puede mejorar la calidad de la experiencia en MR, aumentando el factor de presencia.

2 Definición de proyecto

Este trabajo se basa en investigaciones previas de segmentación egocéntrica en MR, generando un avatar basado en vídeo en tiempo real usando datos RGB-D, y usando la profundidad para crear una representación 3D del usuario en un entorno de MR.

Se han cumplido todos los objetivos primarios y secundarios, lo que ha llevado a aportaciones técnicas y experimentales al campo. Los objetivos se pueden resumir en los siguientes:

- Explorar un nuevo sensor (RealSense D435) que capta frames de color y profundidad. Incluye la calibración de ambas cámaras y la integración física de la cámara en las gafas de MR.
- Actualizar datasets existentes para incluir frames de profundidad.

- Entrenar arquitecturas de aprendizaje profundo que funcionen en tiempo real capaces de trabajar con datos RGB y RGB-D (Thundernet y AsymFormer).
- Diseñar y desarrollar una aplicación de MR en Unity capaz de renderizar el cuerpo del usuario como una nube de puntos 3D y de comunicarse con un servidor de segmentación desplegado.
- Desarrollar una evaluación subjetiva completa para analizar el impacto de la profundidad.

Estos objetivos se han trasladado en 4 aportaciones técnicas principales:

- Un algoritmo multimodal capaz de segmentar cuerpos egocéntricos basado en la arquitectura de AsymFormer [1]. Hasta donde sabemos, esta es la primera red capaz de segmentar cuerpos egocéntricos usando datos RGB-D como input.
- Creación de un dataset RGB-D para segmentación semántica egocéntrica, como extensión del Egocentric Bodies Dataset [3].
- Integración de un algoritmo de ML en un entorno de MR basado en Unity.
- Una doble evaluación - cualitativa y cuantitativa- que demuestra cómo afecta la profundidad (por separado) a la calidad de la segmentación y a la percepción de usuario en el entorno.

3 Descripción del sistema

Para determinar cómo afecta la profundidad a la segmentación, se han entrenado dos modelos de segmentación: Thundernet y AsymFormer. Los dos se han entrenado con el Egocentric Bodies Dataset [3], extendido a su versión RGB-D en este trabajo. Thundernet [8] es una red basada en una U-net que usa convoluciones para extraer las features. Por otro lado, AsymFormer [1] es una red más moderna que procesa color y profundidad en paralelo. Esto mejora la segmentación, pero sube el tiempo de inferencia.

Para integrar estos modelos de ML en MR, se ha implementado el flujo descrito en la Figura 1. El cuerpo del usuario y sus alrededores se captan con una RealSense D435 unida a las gafas con una carcasa propia impresa en 3D. Unity manda los frames a un modelo de segmentación (AsymFormer o Thundernet). El modelo devuelve una máscara con el cuerpo del usuario, que Unity renderiza como una nube de puntos 3D o una imagen 2D en MR, dependiendo de qué experiencia se seleccione.

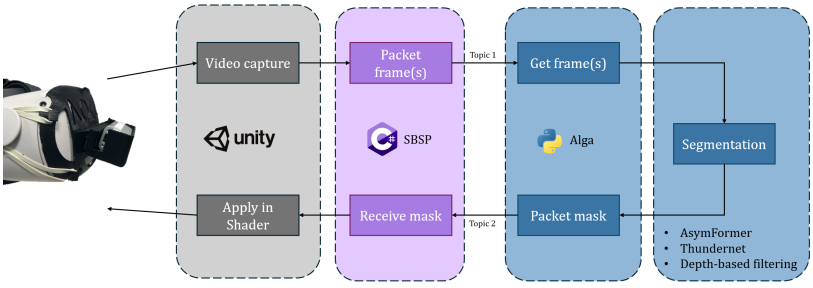


Figure 1: Flujo E2E completo que sigue la aplicación final.

El efecto de usar la profundidad en MR se ha evaluado objetiva y subjetivamente. Se han diseñado dos experimentos. El primero es para determinar si incorporar información de profundidad mejora la calidad de la segmentación y si merece la pena usar una red compleja, que reduce la eficiencia, para mejorar la experiencia en su conjunto. Es una evaluación basada en vídeo en la que se presenta a los usuarios varios resultados de segmentación y tienen que decidir cuál es mejor. Estos resultados se van a comparar con el mIoU medido en frames de los mismos vídeos. La segunda evaluación estudia si representar al usuario

como una nube de puntos (usando profundidad) mejora el factor de presencia respecto a una visualización 2D estándar o no. Este segundo experimento está dentro de un entorno de MR donde los usuarios van a interactuar con un entorno virtual. Su cuerpo se va a segmentar con tres modelos y va a haber dos visualizaciones posibles: una en 2D y otra como una nube de puntos 3D. Los usuarios evaluarán si la profundidad mejora la experiencia global, a pesar de que una nube de puntos no es perfecta en sí. Los modelos que se van a comparar son Thundernet, AsymFormer y un filtro de profundidad (sin ML).

4 Resultados

Sobre la precisión de la segmentación, AsymFormer supera a Thundernet en, aproximadamente, un 3% en el conjunto de test del Egocentric Bodies Dataset, mostrando los beneficios de usar la profundidad para segmentar. Esta mejora se atribuye principalmente a que la profundidad ayuda a descartar falsos positivos en el fondo y a segmentar en zonas con colores ambiguos.

En la evaluación de vídeos, AsymFormer ha liderado subjetivamente, con un MOS de 4.23/5, y objetivamente, con un IoU de 0.82, usando como ground truth etiquetas generadas con SAM [4].

En la evaluación en Unity, los resultados varían. A pesar de que la nube de puntos mejora la percepción de distancias, el modelo mejor valorado ha sido Thundernet con una representación 2D. Según los comentarios de los participantes, esto se debe a las imperfecciones de la nube de puntos en sí, y por problemas técnicos como el FOV limitado de la RealSense, la baja resolución de la cámara y problemas de latencia, sobre todo con AsymFormer. A pesar de esto, el filtro de profundidad con una nube de puntos se ha posicionado como un buen candidato, porque te proporciona sensación de distancias sin ningún tipo de retraso ni falsos positivos cambiantes. Algunos usuarios prefirieron este modelo, destacando su capacidad para interactuar con una escena virtual en 3D. Todos los resultados están resumidos en la Tabla 1.

Model	**EgoBodies dataset**		**Evaluación vídeos**		**Evaluación MR**	
	Test IoU	Velocidad (s)	IoU	MOS	MOS (Global)	MOS (Distancias)
Thundernet Mono	0.895	0.0199	0.72	3.46	–	–
Thundernet Stereo	0.814	0.0199	0.68	3.03	**3.91**	4.25
Filtro de profundidad	–	0.0000	0.32	2.52	3.55	**4.73**
AsymFormer (RGB-D)	**0.926**	0.0329	**0.82**	**4.23**	3.02	4.60

Table 1: Resumen del rendimiento de los modelos en el dataset, en la evaluación de vídeo y la evaluación de Unity (global y percepción de distancias).

5 Conclusiones

Este proyecto demuestra que el uso de la profundidad para segmentación semántica egocéntrica mejora notablemente la calidad de la segmentación y la experiencia inmersiva en aplicaciones de MR. La representación como nube de puntos en 3D mejora la percepción de distancias, sobre todo en escenarios en los que hay que interactuar con la escena. A pesar de esto, las limitaciones en las cámaras de profundidad actuales, como la baja resolución, afectan a la calidad de la reconstrucción en 3D. Además, el alto coste computacional de modelos complejos como AsymFormer hacen que su uso no sea lo más óptimo. Para experiencias sencillas, en las que no hace falta interactuar con el entorno, una representación en 2D más simple sería suficiente. Trabajos futuros podrían explorar representaciones 3D que fueran más estables visualmente y optimizaciones para poder usar modelos más complejos. Mejoras en la tecnología, como sensores de profundidad con un mayor FOV, pueden hacer que aumente aun más la utilidad de la segmentación con profundidad en entornos de MR, permitiendo construir avatares aun más realistas.

References

[1] Siqi Du, Weixi Wang, Renzhong Guo, Ruisheng Wang, and Shengjun Tang. Asymformer: Asymmetrical cross-modal representation learning for mobile platform real-time rgb-d semantic segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7608–7615, 2024. doi: 10.1109/CVPRW63382.2024.00756.

[2] Ester González-Sosa, Guillermo Robledo, Diego González Morín, Pablo Perez-Garcia, and Álvaro Villegas. Real time egocentric object segmentation: Thu-read labeling and benchmarking results. *ArXiv*, abs/2106.04957, 2021. URL `https://api.semanticscholar.org/CorpusID:235377314`.

[3] Ester González-Sosa, Andrija Gajic, Diego González Morín, Guillermo Robledo, Pablo Pérez, and Álvaro Villegas. Real time egocentric segmentation for video-self avatar in mixed reality. *ArXiv*, abs/2207.01296, 2022. URL `https://api.semanticscholar.org/CorpusID:250264412`.

[4] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[5] Kwan Min Lee. Presence, explicated. *Communication Theory*, 14(1):27–50, 2004. doi: 10.1111/j.1468-2885.2004.tb00302.x. URL `https://academic.oup.com/ct/article/14/1/27/4110793`.

[6] Diego Gonzalez Morin, Ester Gonzalez-Sosa, Pablo Perez, and Alvaro Villegas. Full body video-based self-avatars for mixed reality: from e2e system to user study. *Virtual Reality*, 27(3):2129–2147, 2023. URL `https://doi.org/10.1007/s10055-023-00785-0`.

[7] Richard Skarbez, Missie Smith, and Mary C Whitton. Revisiting milgram and kishino's reality-virtuality continuum. *Frontiers in Virtual Reality*, 2:647997, 2021.

[8] Wei Xiang, Hongda Mao, and Vassilis Athitsos. Thundernet: A turbo unified network for real-time semantic segmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1789–1796, 2019. doi: 10.1109/WACV.2019.00195.

Contents

List of Figures

List of Tables

Using Depth for Enhancing Semantic Segmentation for Extended Reality Applications

Sofía Pedrós Tobaruela

Abstract

Extended Reality (XR) technologies offer immersive experiences blending virtual and real environments. For an optimal experience, users must feel integrated in the scene. Therefore, an accurate depiction of the user should be incorporated into the virtual environment. This is often achieved by means of video-based self-avatars created via egocentric semantic segmentation. The present work evaluates the integration of depth to create these self-avatars to improve the representation of the user's body, thereby enhancing the user's perception within the XR application. More specifically, we introduce a real-time depth-enhanced semantic segmentation model, trained with a RGB-D egocentric dataset, and an approach where the user's video-based self-avatar is reconstructed as a 3D point cloud in the final application. We show that using depth improves segmentation performance, with a 13.75% relative increase in mIoU and a 39.6% gain in subjective quality, and that representing the user as a 3D point cloud enhances distance perception in XR compared to previous implementations that do not include depth.

1 Introduction

Egocentric vision refers to the field of computer vision which analyzes images from a first-person point of view (POV), typically captured with wearable vision systems, such as head-mounted displays (HMD), including virtual reality headsets. This perspective is particularly relevant in applications that aim to understand human activity from the user's visual field. Egocentric vision has proven useful across various contexts. The most prominent are: (i) life logging, which captures and analyzes daily activities; (ii) action recognition, mainly used in robotics; and (iii) scene analysis.

In recent years, egocentric vision has become increasingly relevant in Extended Reality (XR) systems, which are often designed from a first-person POV. In particular, Mixed Reality (MR) blends the virtual and real environments, creating a highly immersive experience [24]. In this context, egocentric data can serve as a valuable input to enhance the so-called self-perception [16]. This is a critical aspect of immersion in virtual environments that reflects the user's perceived integration within their virtual surroundings. Prior research has aimed to enhance self-perception with the use of avatars, (i.e digital representations of the user), demonstrating their potential to improve immersion [2, 28]. In this context, the use of egocentric vision can serve as an advantage to create realistic avatars, as proposed by Morin et al. [19], as egocentric images can be used to create a video-based self-avatars that can be integrated in an XR application.

One approach to generating these realistic video-based self-avatars is egocentric semantic segmentation. This process involves segmenting the user's body from a first-person POV and then rendering it in the virtual environment. This has been previously used in MR applications, as seen in various implementations [19]. However, for this machine learning (ML) approach to work in MR, the segmentation algorithm must: (i) be robust under uncontrolled conditions; and (ii) achieve real-time performance.

ML solutions have already achieved remarkable results [6, 8, 19]. For instance, González-Sosa et al. [8] proposed a model, Thundernet, capable of performing real-time inference in MR using stereo-captured RGB images. Despite these advancements, the displayed 2D visualization did not provide an accurate perception of distances. Additionally, background objects with skin-like tones would frequently be classified as false positives (segmentation errors). These errors suggest that depth information may improve segmentation quality and reduce the rate of false positives, thereby increasing the presence factor.

The proposed work extends the previous implementations with the following contributions:

- A multimodal algorithm capable of segmenting egocentric bodies based on the AsymFormer architecture [4]. To the best of our knowledge, this is the first network able to segment egocentric bodies in real time using RGB-D data.
- Creation of an RGB-D dataset for egocentric semantic segmentation, through the extension of the Egocentric Bodies Dataset [8].
- Integration of the ML algorithm in a MR environment based on Unity.
- A double evaluation that measures, both qualitatively and quantitatively, how depth affects, on one hand, the segmentation performance, and on the other, the user's perception within the environment.

This work is structured as follows: section 2 reviews related work and section 3 explains the methodology that will be followed; section 4 offers a detailed description of the ML models that will be used; section 5 presents the RGB-D dataset used for the task; section 6 provides details on the system design. Section 7 presents the evaluation process and section 8 discusses the obtained results. Finally, section 9 concludes the paper with final remarks and future research lines.

2 Related Work

The development of technologies for effective egocentric segmentation techniques for MR has seen significant progress in recent years. This section will highlight the key previous research that served as motivation for this work, organized around four main areas: RGB-D data, real time semantic segmentation, semantic segmentation with RGB-D data and egocentric semantic segmentation.

2.1 RGB-D Data

Recently, it has been seen that the use of depth images to complement RGB data can enhance the performance in different computer vision tasks. It has been particularly useful in action recognition in indoor environments, where depth data enhances the scene understanding, making easier for the model to differentiate between classes in color-ambiguous environments. Although RGB-D sensors introduce challenges such as noise and limited range, their ability to capture 3D structure has proven advantageous for tasks requiring spatial reasoning [17, 22].

A multimodal input is specially beneficial in tasks with a dynamic environment or where the number of existing datasets is limited [17], which applies to egocentric segmentation. For scene analysis, it has been concluded that RGB-D data provides the richest information to optimize efficiency and effectiveness as it combines 2D and 3D data [4].

Depth is usually acquired through triangulation and Time-of-Flight techniques. Across these techniques, the most used are the ones reliant upon structured light (IR and LiDAR) [22]. The growing use of portable sensors such as the RealSense, Microsoft Kinect or the Orbbec [22] allow for depth information to be captured even in egocentric scenarios.

2.2 Real time semantic segmentation

Real-time semantic segmentation focuses on achieving a balance between accuracy and inference speed. To satisfy this real-time constraint there are two main approaches: traditional (non-ML) and with ML, with the latter offering consistently better results. Non-real time segmentation solutions were out of the scope for this thesis.

Early non-deep learning approaches primarily involved using depth [15] or color [3] as thresholds. The idea is simple: keep the parts of the image with a certain color or up to a specific depth. While straightforward, these techniques present certain drawbacks. Color-based segmentation struggles with objects that have skin-like colors and varying skin tones. Depth-based segmentation problems arise from the use of sensors that tend to be noisy and have a narrow field of view [6].

In contrast, deep learning has emerged as a promising candidate for the task. Gonzalez-Sosa et al. [6], among other authors, have shown that deep learning architectures tend to have more accurate and consistent results. A model is considered real-time if it is be able to process a minimum of 30 frames per second (FPS). To achieve (or surpass) this benchmark, models should leverage some GPU-friendly techniques. Table 1 offers a comprehensive review of state-of-the-art models showing different strategies that have been used to find a balance between accuracy and speed with RGB images.

Model	**Performance (% mIoU)**		**Speed (FPS)**		**Idea**
	Cityscapes (512×1024)	CamVid (720×960)	Cityscapes (512×1024)	CamVid (720×960)	
ENet	58.3	51,3	76.9	61.2	Architecture designed for tasks requiring low latency operation.
BiSeNet-Res18	74.7	68.7	65.5	116.3	Treat spatial details and categorical semantics separately to increase accuracy.
BiSeNet-Xception39	68.4	65.6	**105**	124.1	BiSeNet using Xception as a backbone.
RTFormer-Base*	**79.3**	**82.5**	39.1	94	Use GPU-friendly operations in attention to use transformers.
RTFormer-Slim*	76.3	81.4	**110**	**190.7**	Lightweight version of RTFormer.
PP-LiteSeg	**77.5**	**75**	102.6	**154.8**	User optimized decoders, attention fusion methods and pooling mechanisms.
STDC-Network	76.8	73.9	97	152.2	Extract deep features removing rebundancy in BiseNet architecture.
LEDNet	70.6	x	71	x	Lightweight Encoder-Decoder Network.
EsNet	70.7	x	62	x	Optimize CNN-based architectures with downsampling, upsampling, factorized convolution units and its parallels.
ThunderNet	64	x	96.2	x	Unifies the pyramid pooling module with a customized decoder. It can achieve **214** FPS in a GTX 1080 Ti. [20]

Table 1: Comparison of semantic segmentation models. Performance measured on an NVIDIA GTX 1080Ti. Models marked with * were evaluated on a different hardware (NVIDIA RTX 2080Ti).

2.3 Semantic segmentation with RGB-D data

Few models support semantic segmentation using RGB-D input, and even less can operate in real-time. However, it is well-established that the use of RGB-D data increases segmentation accuracy [4], even though the variety of models able to so is limited [9]. Two key models are AsymFormer [4] and Efinet [32].

AsymFormer [4] builds on prior research that suggested that the RGB and depth features needed to be processed separately and then combined. Moreover, earlier works had shown that the RGB information tended to be more relevant to the final decision. Despite this, previous models used a symmetrical backbone to process the information. The contribution from the Asymformer model came with three main ideas: (i) using an asymmetrical backbone, with a bigger part to process the RGB branch; (ii) introducing a Local Attention-Guided Feature Selection (LAFS) module to dynamically select the most important features from both branches; (iii) and proposing a Cross-Modal Attention-Guided Feature Correlation Embedding (CMA) module to further enhance multi-modal fusion. This model achieves 54% accuracy on NYUv2 and an inference speed of 65 FPS (79 FPS after quantization) though meeting real-time constraints.

EFINet [32] is a posterior model that challenges the use of a transformer (even a lightweighted one) to process the depth branch, like AsymFormer did. It proposes a lightweight architecture based solely on Convolutional Neural Networks (CNNs) to reduce computational overhead, aiming for real-time performance with only a slight trade-off in accuracy. It achieves 55.5% and 50.6% mean Intersection over Union (mIoU) on the NYUDv2 and SUN RGB-D datasets, respectively, and real-time inference at 31.2 FPS.

2.4 Egocentric semantic segmentation

Egocentric semantic segmentation is a relatively new area that deals with images captured from wearable devices, often in dynamic and cluttered environments [18]. Although most existing methods rely on RGB inputs, incorporating depth information in egocentric settings could be highly beneficial [17].

There are relatively few datasets designed for egocentric semantic segmentation, especially ones where the arms are also segmented as the "*user*" class. Segmenting the arms can be beneficial as they are naturally visible without the headset, though seeing them in the virtual world improves the presence factor [6]. To that end, Gonzalez-Sosa et al. [6] created EgoArms, a pseudo-synthetic dataset designed specifically for segmenting human bodies from an egocentric perspective. Additionally, the same research group introduced EgoBodies [7, 8], another dataset for segmenting the human body.

Building upon these efforts, the aim of this project is to obtain a real-time egocentric semantic segmentation algorithm, capable of working with RGB-D data. By integrating depth into the segmentation and visualization pipeline, we address the problems presented in the introduction of RGB-only approaches.

3 Methodology

This work aims to investigate how depth information can improve MR experiences. Unlike previous works which discard this information, we use depth to improve both, segmentation accuracy and spatial perception. The user's image will be captured with a sensor that captures both color and depth information (RealSense D435).This data will be then processed to construct a 3D point cloud representation of the user, displayed in real-time in MR.

To capture an accurate representation of the user, the sensor will be physically attached to the Virtual Reality headset (Quest 2) via a custom 3D-printed camera cover. Not only will the captured RGB-D data be used for visualization, but also this study will explore how semantic segmentation models can include depth information to increase their accuracy (measured by Intersection over Union - IoU) while

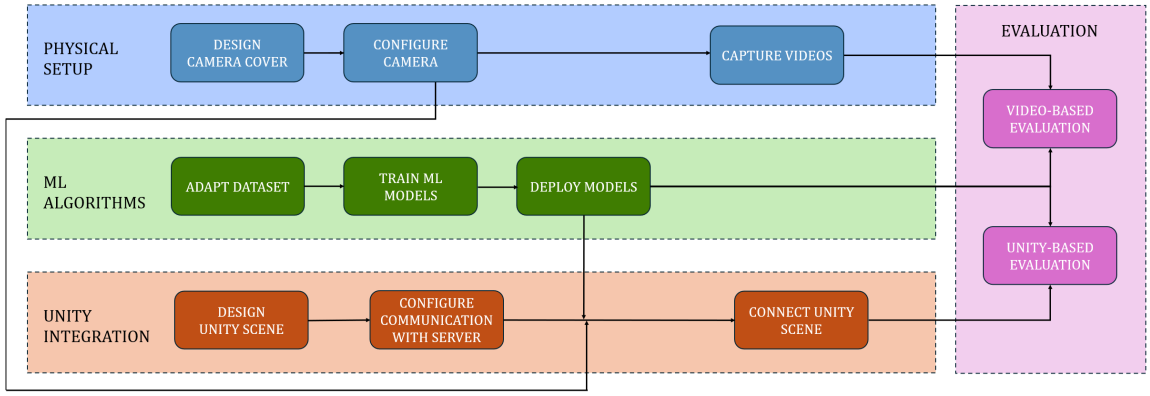


Figure 1: Block diagram representing the steps to be taken to achieve the project's objectives.

maintaining real time performance (measured in frames per second -FPS- in inference time). Finally, various settings will be compared to determine which approach provides a better experience for the end user, by measuring the Mean Opinion Score (MOS) in two experiments, described in Section 7.

With that, the primary and secondary objectives can be summarized in:

- Calibrate the RGB and depth streams of the RealSense D435 and physically attach the device to the MR headset via a custom 3D-printed mount.
- Update existing egocentric segmentation datasets to support monocular RGB-D input, including depth alignment and stereo-to-mono conversion.
- Train real-time semantic segmentation deep learning architectures with RGB data and RGB-D data (Thundernet and AsymFormer).
- Design and develop a Unity-based application capable of rendering the segmented user as a 3D point cloud, including the communication with a deployed segmentation server.
- Conduct a subjective evaluation to assess user perception with and without depth information.

Figure 1 details the steps taken to achieve those objectives. More detail on the steps taken can be found on its corresponding section.

3.1 Hardware and software specifications

The egocentric RGB-D data was captured using an Intel RealSense D435 camera (FOV: 69° × 42°) [10] mounted on a custom 3D-printed cover attached to a Meta Quest 2 headset, and the stereo images were captured with the ELP 960P HD OV9750 stereo camera (FOV: 90º) [5]. The custom 3D camera cover was designed using the online version of TinkerCAD and printed with an Anycubic Kobra 3 using PLA.

AsymFormer was trained with an NVIDIA RTX 4080 GPU using PyTorch 2.5.1+cu118 and Python 3.11.8. Thundernet models were trained using Keras 2.2.4 with Python 3.5 on a dual NVIDIA GTX 1080 Ti (12 GB each) setup. For the experiments, the models were runing on a NVDIA RTX 4080 and an NVIDIA RTX 3070 respectively. More details can be found on Section 7.

4 Deep learning networks

The introduction of deep learning architectures for semantic segmentation has significantly improved the overall segmentation results. The most adopted structure for this this task is the UNet architecture, which was originally proposed for biomedical image segmentation [21], and built upon the fully convolutional

networks, introduced by Shelhamer et al. [23]. These models use convolutional neural networks (CNNs) to reduce the dimensionality of the input and extract key features and then use the inverse operation to return to the original input size and give a prediction for each pixel.

The models based on a UNet architecture have two main parts: an encoder and a decoder. The encoder reduces dimensionality to extract a compact feature representation and then the decoder restores the original dimension to output the segmentation result. Models based on a UNet are often enhanced using techniques like pyramid pooling to capture multi-scale features or some alternative-convolution types to increase efficiency.

Parallel to the development and improvement of CNN-based architectures, the transformer was presented in 2017 [27]. It revolutionized the development of ML models, and was specially useful for Natural Language Processing (NLP) tasks. The extraordinary growth of NLP due to the use of transformers led to its adoption in vision tasks, including semantic segmentation, where they outperformed some CNNs in terms of accuracy. However, the computational complexity of transformers forces models to adapt the attention mechanism to make it GPU-friendly in order to function in real-time. This implies that the number of transformer-based architectures for real-time segmentation is limited.

In this work, we compare the performance of two models—ThunderNet (CNN-based) and AsymFormer (transformer-based)—whose architectures are described in detail below.

4.1 Asymformer

Asymformer [4] is a model that processes RGB-D images to perform semantic segmentation in real time. The motivation behind this model arose from the well-established fact that the use of RGB-D data increases the accuracy of semantic segmentation. However, preexisting models were not capable of performing inference in real time when processing RGB-D data. Moreover, prior research suggested that the RGB and depth features needed to be processed separately and then combined to achieve the best results. Additionally, previous works showed that the RGB information tended to be more relevant to the final decision. Despite this, previous models used a symmetrical backbone to process the RGB-D information. Furthermore, the performance usually depended on how the information from both branches was merged, but no approach has been proven to be universally superior.

The contribution from the Asymformer model consists of three main ideas: (i) using an asymmetrical backbone to process the RGB and depth information, with a larger part to process the RGB branch; (ii) introducing a Local Attention-Guided Feature Selection (LAFS) module to dynamically select the most important features from both branches; and (iii) proposing a Cross-Modal Attention-Guided Feature Correlation Embedding (CMA) module to further enhance multi-modal fusion (combining the information from both branches efficiently). This module computes self-similarity between the different features. Additionally, the model employs a MLP decoder at the end of the architecture to decode the information and produce the final output. The architecture of the model can be found on Figure 2. Each component is described in detailed bellow.

The overall framework consists of two branches: one for the RGB information and one for the depth information. In the RGB branch, since CNNs are usually faster than transformers and this is the "larger" branch, Asymformer uses multiple ConvNext modules (a hardware friendly convolutional network) to efficiently extract key features. The input for each module is the result of the previous module. Between modules, the extracted features are also forwarded to the LAFS and CMA modules to be combined with the depth features. However, this information does not feed back into the RGB branch.

In contrast, as the depth branch is more lightweight and transformers tend to achieve better results in this context, Asymformer utilizes a Mix-Transformer (light-weight efficient transformer). Each module in this branch takes as input the output of the LAFS and CMA modules, except for the first and second modules, which receive the output of the previous module.

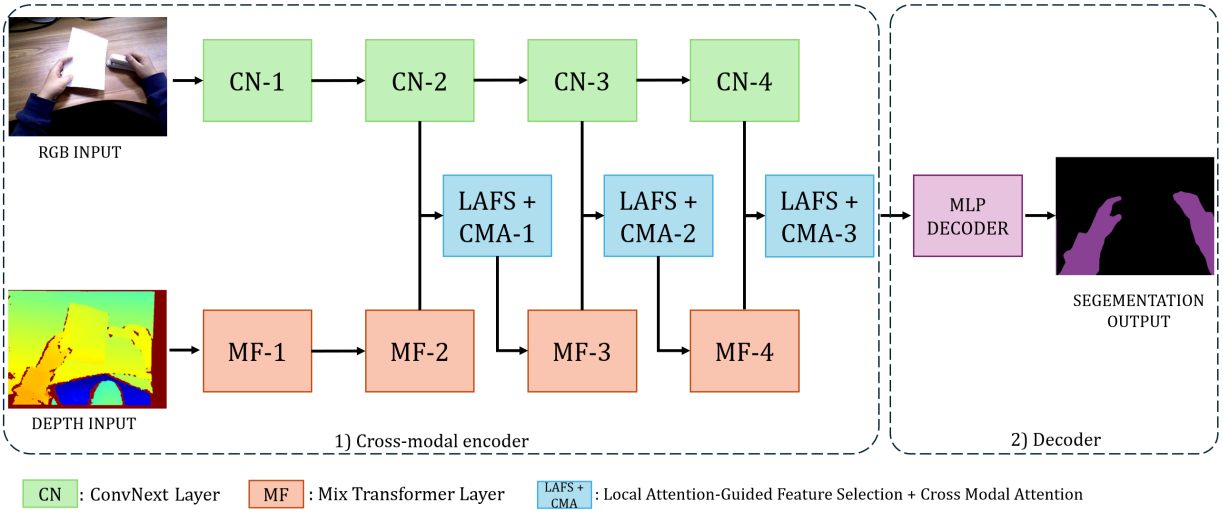


Figure 2: Asymformer complete architecture [4]

.

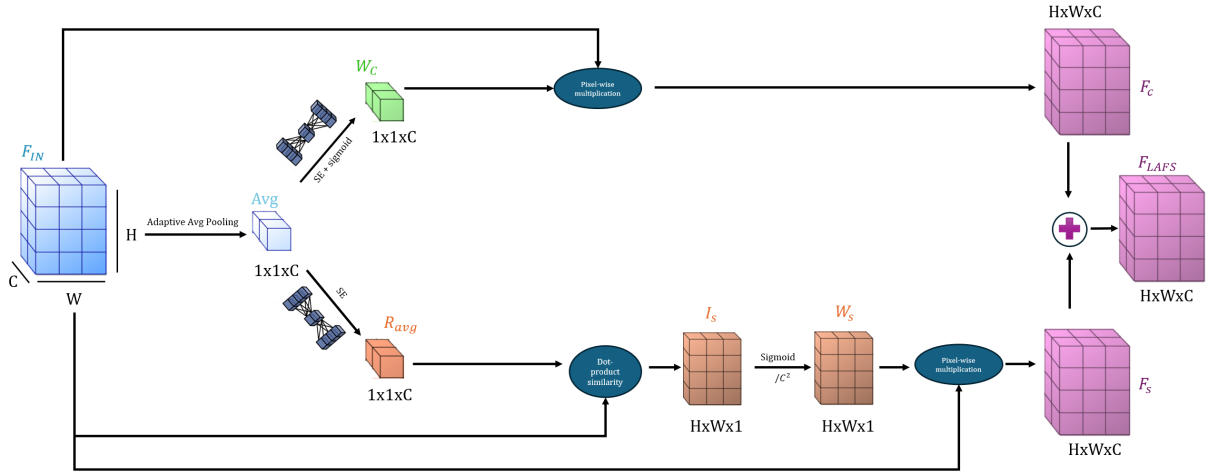


Figure 3: LAFS module in detail.

The **LAFS** module can be found on Figure 3. This module is a learnable approach to compress feature information, in other words, it selects key features from both branches. It is in essence an attention variation, as it has been demonstrated that attention mechanisms can select complementary features from depth and color features, thereby improving the efficiency of feature extraction and the performance of semantic segmentation overall. The LAFS module receives the depth and RGB features concatenated and it returns a weighted result of the inputs. The weights are computed as channel-wise and spatial attention weights from the combined RGB and depth features, allowing the network to focus on the most relevant multimodal information for segmentation.

In contrast, the **CMA** module defines cross-modal self-similarity to determine how the features of one branch interact with the other. The complete outline can be found on Figure 4. The idea is that, rather than just select existing features, extract new information from the fused features. CMA uses a lineal sum to compute this similarity, then, the result is added to the fused features (the output of the LAFS module). By doing this, it enables richer and more discriminative representations through channel-wise information

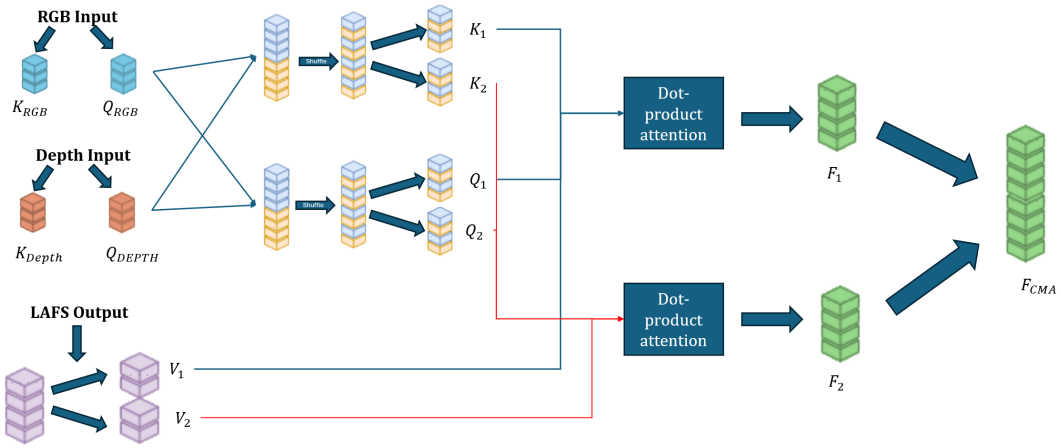


Figure 4: CMA module in detail.

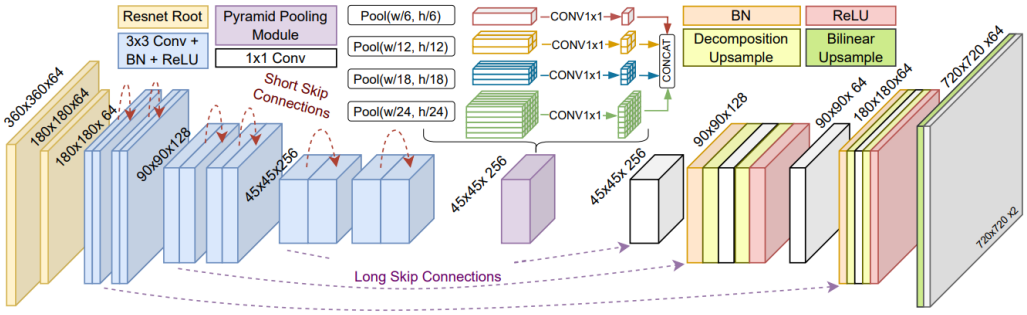


Figure 5: Thundernet architecture.

exchange. At the end, this module is a variation of self attention that computes the similarity between the branches (cross-modal self-similarity).

At the end of the global architecture, the information from the last LAFS + CMA module is passed through an MLP network to output the final segmentation result. A step by step computation of these two modules is presented on Appendix A.

4.2 Thundernet

Thundernet is a model proposed by Xiang et al. [29], that achieves better (faster and more accurate) results compared to previous architectures. This work will use a shallow architecture based on Thundernet, as explored by González-Sosa et al. [8]. The architecture of the proposed model can be found on Figure 5.

This architecture follows a U-Net structure, as it has 3 main parts: (i) an encoder; (ii) a pyramid pooling module; and (iii) a decoder. The encoder extracts key features from the input image, this encoder is formed by 3 Resnet-18 blocks, similar to the original architecture. Then, there is a pyramid pooling module to capture features from multiple scales. The difference here with the original architecture is the use of bigger pooling factors (as the input images are bigger). Finally, there is decoder, that restores the original resolution, formed by 2 deconvolutional blocks. Additionally, González-Sosa et al. [8] introduced

long-term residual connections between the encoder and decoder to try to mitigate vanishing gradients. These connections will be included in the final implementation.

5 Egocentric RGB-D Dataset

The baseline for this work is a version of Thundernet [29] trained on RGB stereo egocentric images. This model was able to perform real-time inference segmenting the user's body from a capture of a RealSense stereo camera and displaying the 2D result into a 3D virtual environment.

The dataset used to train this model is the "Egocentric Bodies Dataset" (EgoBodies), a dataset introduced by González-Sosa et al. [8] to address the lack of datasets for egocentric semantic segmentation. The "Egocentric Bodies Dataset" merges three datasets (EgoHuman, THU-READ and EgoOffices) labeled by González-Sosa et al. [8], and two of which were captured by the same research team. This dataset originally consists of a set of monocular videos that include RGB, IMU and depth information. The RGB videos were sampled and then labeled using Amazon Mechanical Turk (AMT). The resulting dataset consisted on 8873 480x640 images with their corresponding labels. The depth information was captured either with a RealSense D435 (through disparity) or a RealSense L515 (through LIDAR technology), but these depth frames were not associated with the corresponding RGB frames. To train Thundernet to work with stereo images, the images were synthetically duplicated to mimic an stereo capture, and the resolution changed to 480x1280.

In this work we present **"RGB-D Egocentric Bodies Dataset"**, an extension of the original "Egocentric Bodies Dataset" where each RGB frame is paired with an associated depth frame. For THU-READ [26] [25], the corresponding depth videos were sampled to extract the exact same RGB frame labeled before. For EgoHuman and EgoOffices, the procedure was similar, there were associated depth videos captured with the RealSense D435 and the RealSense L515 that were resampled. The depth frames from the L515 had a different resolution and had to be aligned to match the dataset.

Furthermore, 1257 images from the EgoHuman dataset did not have a depth video associated (as they were RGB pseudo-synthetic images). To address this, the depth maps were created synthetically. For this purpose, we used Depth-Anything [30] [31], a model for estimating depth from monocular images. The creators of Depth-Anything did not propose a novel architecture, but rather emphasized the importance of a powerful dataset. The architecture of the model is based on a Vision Transformer (ViT) backbone, trained on an extensive unlabeled dataset of approximately 62 million images. This large-scale training allows the model to achieve state-of-the-art results and an excellent zero-shot depth estimation capability in a variety of settings. These synthetic depth maps will serve as replacements for the missing depth maps in the dataset. Additionally, to evaluate the difference between real and synthetic images, Asymformer will be trained on two sets: one fully-synthetic depth dataset and the other partially synthetic. The partially synthetic dataset will be the complete dataset where the 1257 missing depth maps are replaced by synthetic ones, and the fully synthetic consists on 8005 RGB-D images, where the depth maps were estimated with Depth-Anything. This comparison will be useful to evaluate the influence of realistic depth maps in the model's performance.

To further enrich the dataset, EgoBodies was combined with EPSILON. This dataset was captured, labeled and adapted following the same procedure as EgoBodies [7].

The complete dataset consists on 8005 monocular images for training and 1069 for validation with a resolution of 480x640 (height x width). Figure 6 illustrates sample images of the complete dataset.

6 System Design

The final product of this project is a Unity application capable of plotting the user's body as a 3D point cloud and segmenting it using various segmentation algorithms. The user's body and surroundings are

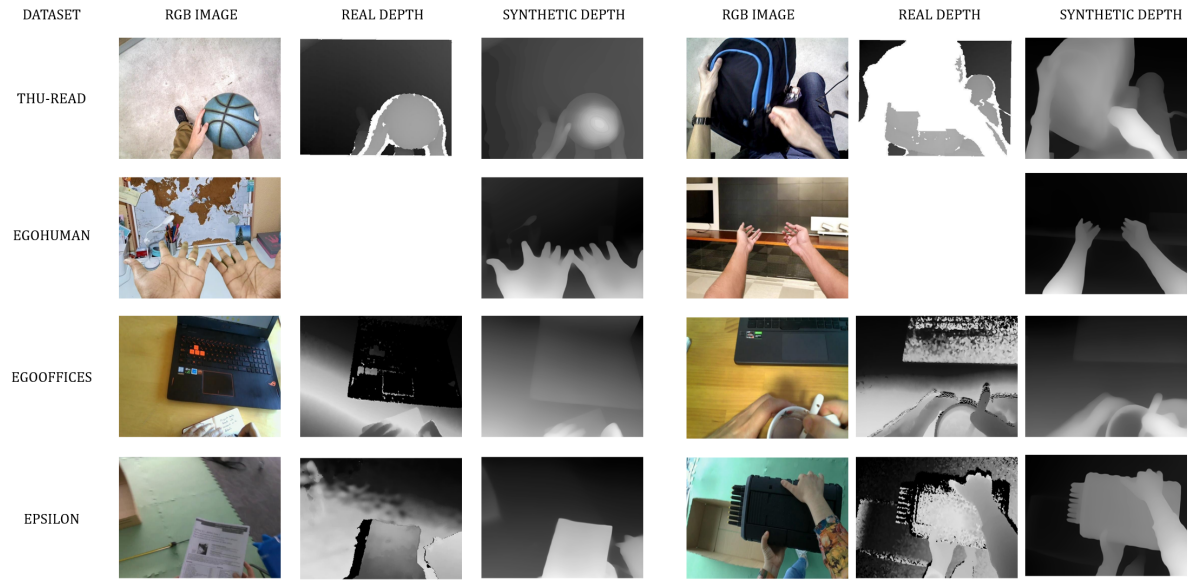


Figure 6: Samples from the RGB-D Egocentric Bodies Dataset. The images from the EgoHuman dataset do not have an associated depth frames as they were RGB images created from a synthetic methods.

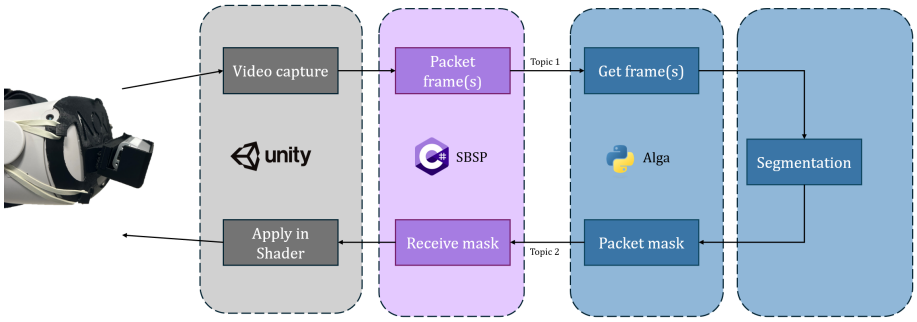


Figure 7: E2E architecture outlined by Morin et al. [19] adapted to the corresponding hardware.

captured with a RealSense D435. This image is sent to a semantic segmentation model via Unity. Finally, the model returns a mask corresponding to the user's body that Unity visualizes as a 3D point cloud. The final end-to-end (E2E) architecture to achieve this can be found on Figure 7. The final Unity application and the model inference will be running in real time in one RTX 4080 GPU.

To ensure the camera captured image and the user's POV are aligned, a custom 3D camera cover was designed to physically attach the RealSense D435 camera with the Quest 2 headset. Since this camera cover introduces an offset between where the data is captured and where the user sees the scene in the headset, this offset was compensated in the final Unity application to align the point cloud correctly.

The path that the data will follow is based on the pipeline introduced by Morin et al. [19]. The detailed path is as follows:

1. A RealSense D435 will capture the RGB and depth information in 30FPS with a resolution of 480x640. The code for the camera capture was designed with C# in Unity using the librealsense plugin for the RealSense [11].

2. Unity will send the RGB (or RGB-D) frame to a server using SBSP. SBSP is a custom data transfer protocol based on ZMQ that allows to send data from Unity to a server and receive data from a server in Unity. It follows a publisher-subscriber structure and uses a TCP protocol. It works with the help of Gstreamer, a library written in C for tasks related to media. In this case, it supports SBSP to work with the RGB and depth frames as well as to receive the mask (an image).
3. A server deployed using Alga receives the data. Alga is another custom library that allows to exchange data between nodes, efficiently handling the data reception and transmission between them using a TCP protocol.
4. The model segments the received image and outputs a segmentation mask. This mask will have a 0 for the background and a 1 for the foreground (the user's body).
5. Alga will send the mask to Unity using a different topic.
6. Unity will receive the mask thanks to SBSP (that will have an instance listening in the corresponding port for an image with the mask's characteristics).
7. A custom shader in Unity will overlay the mask onto the RGB data, rendering the result as a point cloud object, which combines the depth and RGB information. This will display in the Virtual Reality headset a 3D model of the user. Additionally, to reduce false positives, this shader will apply a depth threshold to plot only the points up to a certain distance (2 meters approximately). This threshold can be adjusted if necessary for different users and/or environments.

As it can be appreciated in Figure 7, the data will be sent to the server using a topic and received by Unity using a different topic. The server will have the model deployed awaiting to receive frames. This model can be selected easily, depending on the desired segmentation. For the experiments, to enable baseline comparison, this pipeline was modified (only for the baseline case) by changing the 3D point cloud with the (previous) 2D display, but the data path stayed the same.

7 Evaluation

Before evaluating the performance of the models, we trained multiple real-time semantic segmentation architectures using both unimodal (RGB) and multimodal (RGB-D) inputs from the Egocentric Bodies Dataset. These models were evaluated on the monocular version of the RGB-D Egocentric Bodies Dataset and the results are presented on sections 7.1 and 7.2.

The baseline model that will be used is Thundernet in stereo, which was trained on the stereo version of the Egocentric Bodies Dataset by González-Sosa et al. [7]. It was trained using Keras version 2.2.4, Python 3.5 and two GPU GTX-1080 Ti with 12GB RAM each. The three Resnet-18 blocks at the beginning of the encoder were pre-trained on Imagenet [14] and then finetuned with the EgoBodies Dataset. After an exhaustive grid search of parameters, the version that achieved the highest test mIoU was with the following configuration: learning rate of 1e-4, 20 epochs, weight decay of 2e-4, and batch size of 4 with the Adam optimizer.

The newer version of Thundernet (trained with the monocular version of the dataset) was trained using Keras version 2.6.0 and Python 3.9.10 and a Nvidia GPU 1060. Again, the three Resnet-18 blocks at the begining of the encoder where pre-trained on Imagenet and then finetuned with the egocentric dataset. The best hyperparameters were found with Optuna [1], an automatic hyperparameter optimization software framework. The objective set was to find the highest validation IoU across 100 trials. The final training configuration was: learning rate of 1.37e-4, weight decay of 1.17e-5, batch size of 4, 14 epochs with the Adam optimizer.

AsymFormer was trained with the the torch 2.5.1+cu118 version, Python version 3.11.8 and a Nvidia GPU RTX 4080. Model weights were initialized from a pre-trained checkpoint on the NYUv2 dataset. To search the best hyperparameter configuration, again, Optuna was used, but, in this case, the best

configurations were found with a grid search. The final best model was found with a learning rate of 1e-4, no regularizer, batch size of 1, 14 epochs and the AdamW optimizer. The same training configuration resulted in the highest test IoU for both variants of the AsymFormer model—one using synthetic depth inputs and the other using real depth inputs—facilitating a direct performance comparison between the two.

7.1 Quantitative results

This work has evaluated the objective comparative performance of different real-time semantic segmentation models: (i) Thundernet trained with stereo images (also referred as "baseline"); (ii) Thundernet trained with monocular images; (iii) AsymFormer trained with synthetic depth maps; (iv) AsymFormer trained with a mix of real and synthetic depth maps. The results of the best models can be found on Table 2. This Table compares the mIoU and inference speeds of Thundernet and AsymFormer across the train and test sets. The results demonstrate how the use of depth inputs (in AsymFormer) improves the segmentation results, but it reduces the network efficiency. Nevertheless, both models appear to satisfy real-time constraints. The increase in performance of AsymFormer was up to a 3%, comparing it with the Thundernet Mono, and there was an almost 10% increase in mIoU when comparing AsymFormer with the baseline model (or a relative increase of 13.75%).

Figure 8 illustrates the segmentation outputs across the different models on self-captured videos. From this image, it appears that the better performance of AsymFormer comes from the discard of false positives in the background. It can be seen that Thundernet often classifies objects with skin-like tones as false positives, which are correctly classified as background with AsymFormer. This is likely due to use of depth, which allows the model to accurately identify those objects are too far away to be considered part of the egocentric body.

Model	Data Source	Test IoU	Train IoU	Inference Time (s)	FPS
Thundernet Stereo	RGB stereo images	0.81395	0.8718	0.0199	50.25
Thundernet Mono	RGB monocular images	0.895	0.96	0.0199	50.25
AsymFormer	RGB-D real images	0.9259	0.9344	0.0329	30.39
AsymFormer	RGB-D synthetic images	0.9164	0.9347	0.0329	30.39

Table 2: Comparison of different models and their performance evaluated in the EgoBodies (monocular) dataset.

Comparing both Thundernet models, the model trained on a stereo database achieves a surprisingly high IoU on the monocular dataset, suggesting a strong generalization capability between image sources. Nevertheless, the higher accuracy of the Thundernet Mono can be attributed to the use of Optuna for a more extensive hyperparameters search, possibly leading to a more optimized configuration. This can be justified by the results displayed in Figure 8, which generally show better results for the monocular version, despite the stereo input being adapted for the task. Some examples of this behaviour can be found on Figure 8 A, where the body is segmented better with Thundernet mono or Figure 8 B where the number of false positives is clearly reduced in the monocular version.

It is also noticeable the difference between the AsymFormer model trained only with synthetic data and the one trained with a mixture of real and synthetic images. Even though both of them surpass the quality of Thundernet, the one with real images achieves even better results (which are specially notable with self-captured videos). This could be due to the fact that depth images tend to be noisy, whereas the artificially-generated ones tend to be overly smooth or idealized. This may imply that incorporating real images during training likely helps the model generalize better to the noisy characteristics of real-world depth data.

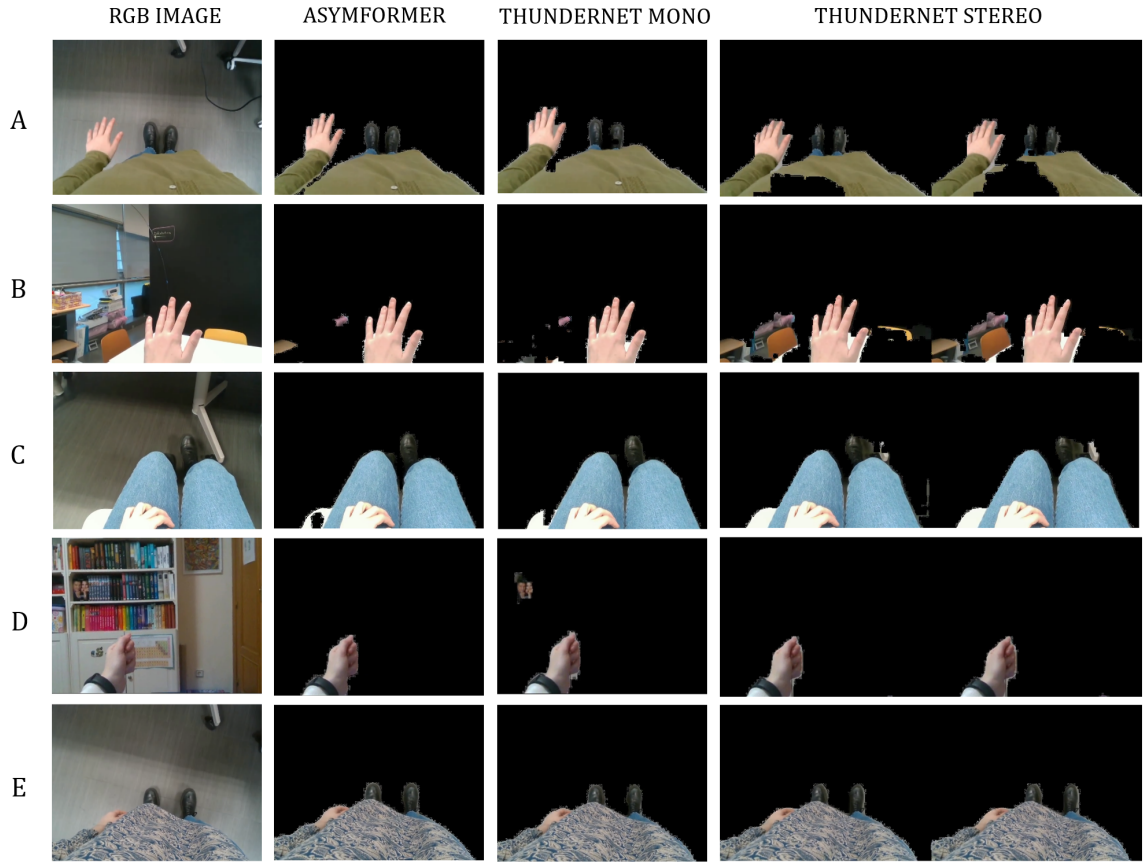


Figure 8: Comparison results of segmentation models on captured videos with the RealSense.

7.2 Qualitative results

The goal of this subjective evaluation is to assess how depth affects the user experience in a MR application. This is particularly relevant because the end users are human and their perception should be considered when studying the self-presence factor.

This study consists of two evaluations. The first one is to determine whether incorporating depth information improves segmentation quality and if the reduction of efficiency that comes with using more complex networks is justified to improve the final experience. The second evaluation examines if representing the user with a point cloud (using depth data) enhances the sense of presence compared to a standard 2D representation or does not provide a significant improvement. To this end, two separate assessment have to be designed: (i) a video-based evaluation where we evaluate the segmentation accuracy of various models; and (ii) a MR-based experience where we evaluate the models integrated in a MR-based experience (models integrated in Unity).

The following research questions were established:

- RQ1: Does the use of depth improve the perceived quality of the egocentric segmentation?
- RQ2: How does distance perception vary using a 3D representation (e.g, point clouds) compared to a 2D visualization?
- RQ3: How does embodiment and self-perception differ between a 3D and a 2D representation of the user?

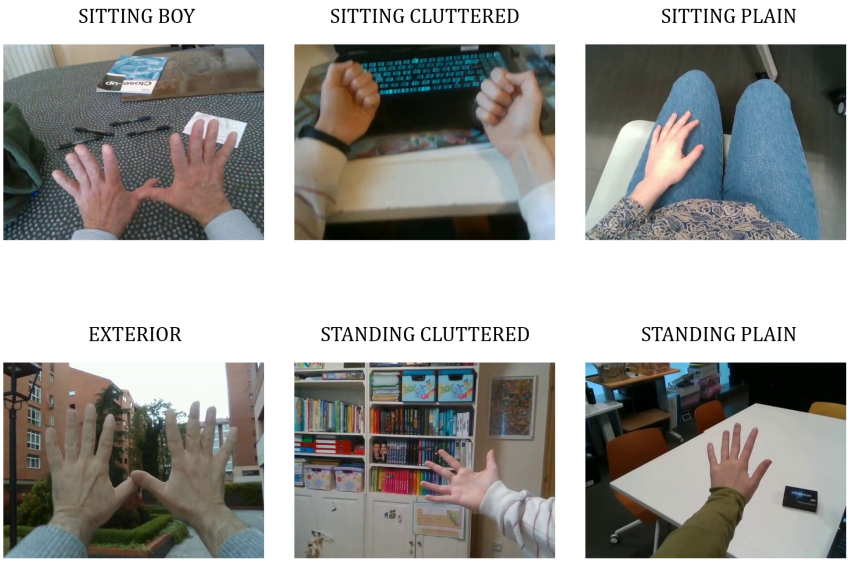


Figure 9: Frames extracted from every setting of the video-based evaluation videos.

7.2.1 Video-based evaluation

This first experiment is a video-based experience to assess the segmentation quality. The users will be presented several segmentation outputs. Users will subjectively evaluate which segmentation they perceive as better in a web platform.

For this evaluation, six short videos were recorded from an egocentric perspective using an Intel RealSense D435 camera, which captures both RGB and depth data. Figure 9 reflects frames extracted from every video in the evaluation with its corresponding description. The scenes cover diverse scenarios: five indoors and one outdoors, with four videos featuring female hands and two featuring male hands. The participants in the videos were standing for three of them and sitting down for the remaining three. To ensure diversity in the backgrounds, three clips had plain backgrounds while the rest included cluttered scenes that could lead to false positives. Each video was segmented using four models: (i) depth-based filtering (discarding points which are 1 meter or further); (ii) AsymFormer; (iii) Thundernet Mono (trained on a monocular dataset); (iv) Thundernet Stereo (used as baseline, trained on a stereo dataset and used in earlier implementations). This resulted in 24 separate videos to assess.

The evaluation was designed using QualityCrowd [12], a framework to perform subjective quality assessment with crowdsourcing. For each case, a 10-second video was presented. This video concatenated side by side the original clip with the segmentation result, as it can be seen in Figure 10. Participants were asked to assess the quality of the segmentation of arms, legs and body (separately) using a 5-point Likert scale (were 1 = bad and 5 = excellent). Additionally, participants were asked if they perceived false positives and how annoying they were, also, with a 5-point scale (were 1 = very annoying and 5 = not perceptible). The evaluation was performed by 20 users.

The results of this subjective evaluation, presented in Figure 12, indicate that AsymFormer outperforms other models in segmentation quality across various contexts for several motives. First, in regard of false positives, AsymFormer was the model where more users declared false positives were not perceptible or not annoying, as showcased in Figure 11. This behavior can be attributed to the use of depth, which, indeed helps the model discard skin-toned like objects in the background. Secondly, as it can be seen in Figure 12a, AsymFormer is the model that can achieve the highest performance in all the settings presented in the videos. The only exception appears in certain cases involving the depth filter, which can be explained by a

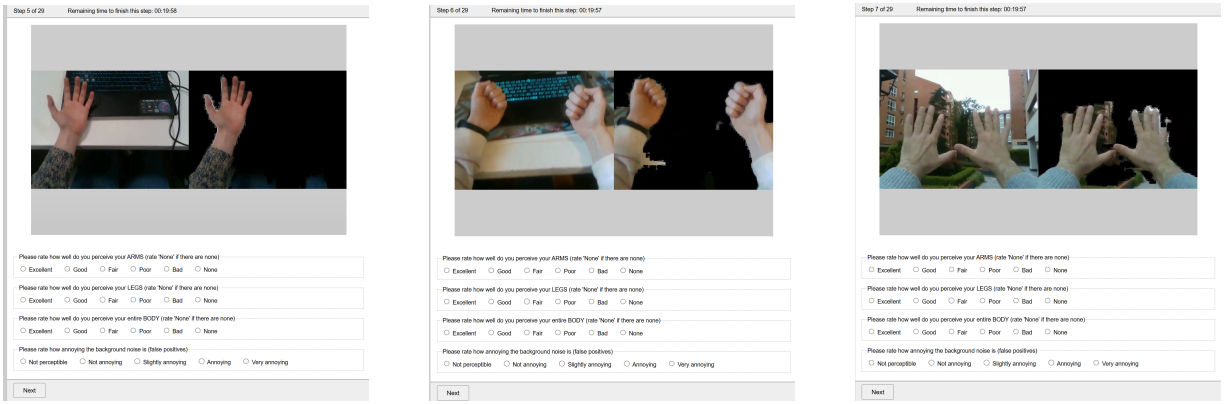


Figure 10: Interface for the video-based evaluation.

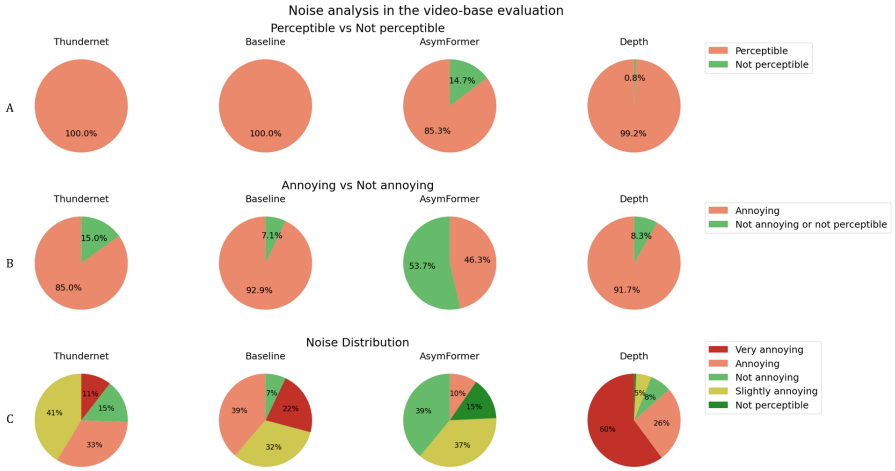
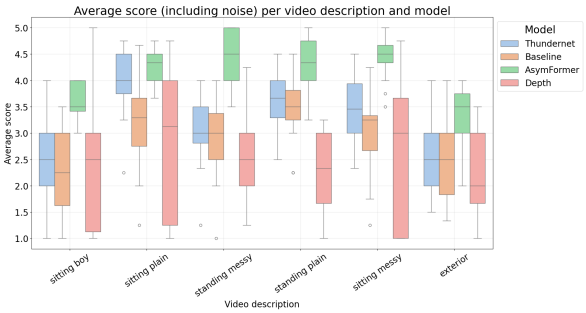


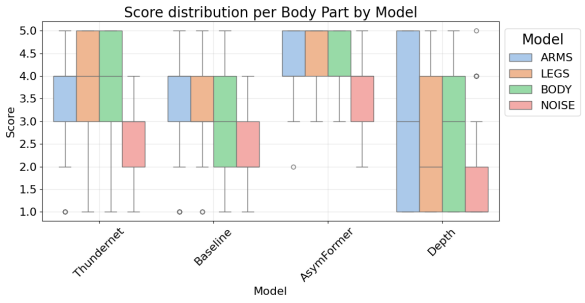
Figure 11: Noise study results across different models in the video-based evaluation.

shift in user interpretation: when all body parts are clearly visible due to the depth filter, some participants focused on visibility rather than segmentation quality. Consequently, a low percentage of responses rated the depth-filtered output as "excellent." Finally, as it can be seen in Figure 12b, AsymFormer achieves the highest scores in every category. Notably is also the model with less variance, proving how the depth filter results may not suffice, depending on the user (as the depth filter reaches the highest and lowest ratings).

To strengthen the significance of these results, an objective measure of the segmentation quality has also been evaluated. Initially, the original videos were sampled and labeled manually using the Segment Anything (SAM) model [13]. The frames were extracted every second, making a total of 10 frames per 10 second video. Subsequently, the masked videos were resampled, and the IoU was computed. Table 3 displays a comparison between the subjective evaluation results and the objective IoU score. The average evaluation score was computed as the mean between the body parts (when perceptible) and the noise (where higher scores are associated with less noise annoyance). As it can be seen, again, the AsymFormer model achieves the highest results on average. The decrease from the train and test IoU score can be partially justified for the ground truth labels that were used, as they were computed with SAM, and results may not be perfect. Nevertheless, as it can be seen in Figure 13, again AsymFormer presents the higher segmentation quality across all situations, with 50% of the results above the 0.8 threshold, and the less variance, which suggest the generalization capability achieved by a more complex model and the usefulness of depth to improve semantic segmentation outputs. This results translate into a relative

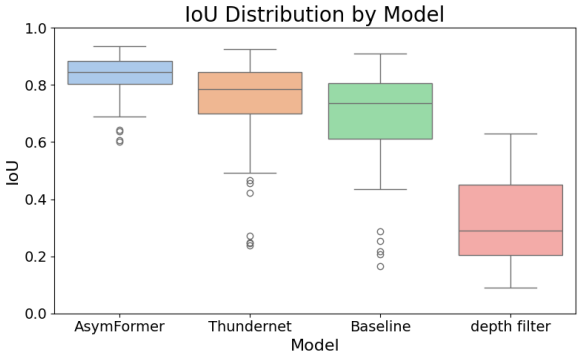


(a) Evaluation scores across different scenarios for each model.

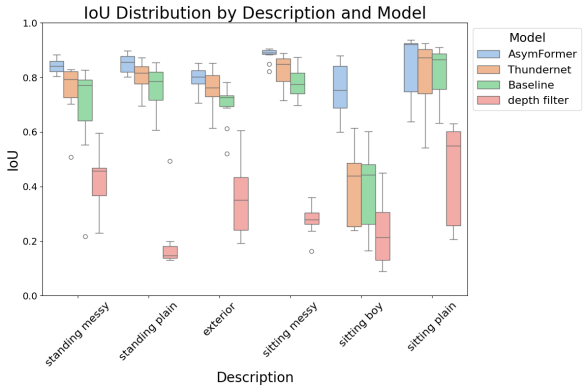


(b) Evaluation scores by body part and noise type; higher values indicate less perceptible false positives.

Figure 12: Subjective evaluation of segmentation performance across models, scenarios, and body regions for the video-based evaluation.



(a) IoU scores per model based on video-based evaluation.



(b) IoU scores by video setting and model.

Figure 13: Objective video-based evaluation of segmentation quality using Intersection over Union (IoU).

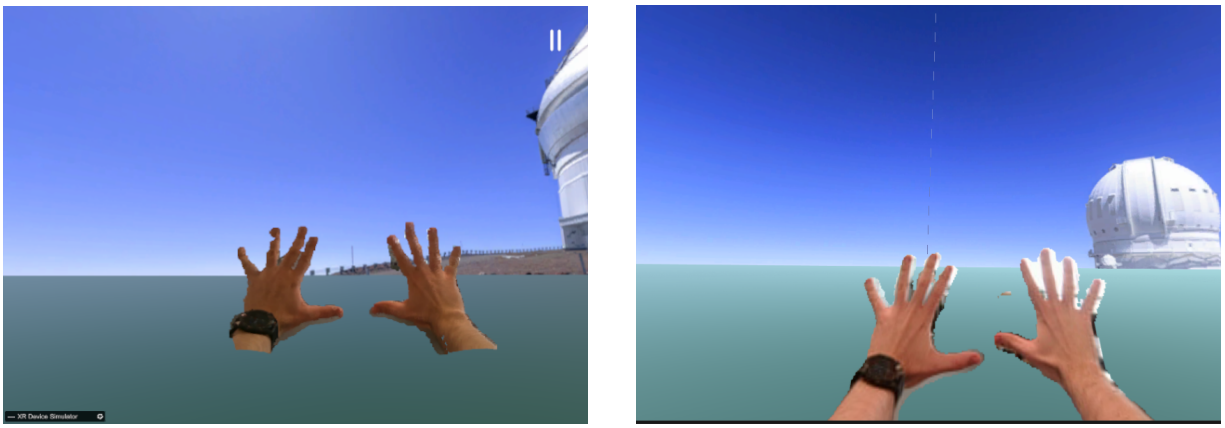
increase in IoU of 20.23% and a relative increase of 39.6% in perceived quality (comparing AsymFormer to the baseline model).

Model	**IoU (0-1)**	**Average Evaluation Score (0-5)**
Thundernet Mono	0.7253	3.46
Thundernet Stereo (Baseline)	0.6885	3.03
AsymFormer	**0.8278**	**4.23**
Depth Filter	0.3213	2.52

Table 3: Comparison of different models and their performance evaluated subjectively and with the IoU for the video evaluation.

7.2.2 MR-based evaluation

The second experiment is embedded within a MR environment, designed in Unity, where the users will interact with a virtual environment. Their body will be segmented using three different models and two visualizations modes will be available: one that showcases the segmentation result as a point cloud (for the depth-filter and AsymFormer) and the other will plot it in 2D (for Thundernet). Users will assess whether the increased depth perception enhances their overall experience, even if the point cloud representation



(a) Unity scene – 3D point cloud segmented by AsymFormer.

(b) Unity scene – 2D view segmented by Thundernet.

Figure 14: Unity scene representations of the user's body using different segmentation and visualization methods.

may flicker or exhibit blind spots due to the depth sensor limitations. This evaluation will be done while the users are in an immersive environment (with a VR headset).

In this environment, the users could see themselves segmented and visualized within a simple virtual scene. This scene had a realistic image in the background and a plain green floor positioned where the ground to avoid "floating" in the environment. Participants could move freely within the space, which was a big room with little to no background objects.

A total of three different scenes were presented to each user: (i) a 3D point cloud segmenting the body with a depth-based filter. The depth threshold for this scene was set to 1.5 meters; (ii) a 3D point cloud segmenting the body with AsymFormer; (iii) a 2D image of the body segmented with the Thundernet Stereo model (corresponding to the baseline system used in previous deployments). The scenes were presented in a random order to avoid possible biases.

The point cloud representations aimed to enhace distance perception, whereas the 2D image served as the control condition. All scenes were rendered in real-time using Unity, with each model running locally on its respective machine. Two different physical setups were used in parallel:

- A computer (RTX 4080) running the AsymFormer model as well as the Unity project (2022 version) running the AsymFormer and depth-filtering scenes. This computer was connected to a pair of Quest2 virtual reality headset, with the RealSense D435 camera attached via a custom camera cover which was designed for this purpose.
- A computer (RTX 3070) running the Thundernet Stereo model as well as the Unity project (2019 version) running the Thundernet scene. This computer was connected to a different set of Quest 2 glasses, with the stereo camera attached via a custom camera cover.

Each evaluation session was conducted individually. Participants could rotate and look freely in all directions to explore their segmented representation. During each session, participants were given an amount of time to freely explore their body in the environment. When they affirmed they were ready, a set of questions was asked by the evaluator (who inserted the answers via an Outlook Forms questionnaire) for each of the setups. The evaluator was the one inserting the answers in the questionnaire while the participants had the VR headset on, to ensure they answer the questions based on their direct experience. The aspects they were asked to assess are the following: (i) perceived quality of arms, legs and body on a 1-5 scale; (ii) perception of delay and whether it was annoying; (iii) accuary of distance perception on a 1-5 scale; (iv); presence and severity of false positives in a 1-5 scale; (v) overall quality of the experience

Model	Overall Score	Arms	Legs	Body	Noise	Delay	Distance Perception	E2E overhead	
								s	FPS
Thundernet Stereo	**3.86**	**4.13**	**4.0**	**3.73**	3.4	4.0	4.26	0.0513	19.5
Depth Filter	3.26	3.4	3.4	2.6	3.2	**4.26**	**4.73**	0.0	–
AsymFormer	2.6	2.2	3.0	2.0	**4.0**	2.8	4.6	0.0756	13.2

Table 4: Comparison of different segmentation models in the MR-based experience on various criteria. Higher scores mean better experiences.

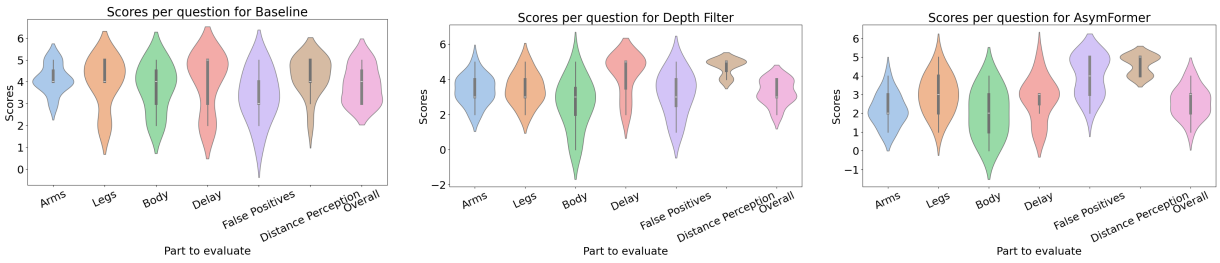
on a 1-5 scale; (vi) any comments they had on the experience, including the criteria they used to evaluate the overall experience. Figure 14 shows examples of the final Unity visualizations that users could see when their body was segmented (with AsymFormer and Thundernet) in MR.

There were 15 participants, 10 males and 5 females, all with technical studies. The average age was 30 with an standard deviation of 12.85 years, and the average height was 172.45 cm with a 10.10 cm standard deviation. Regarding previous VR experience, 6 declared none, 5 little experience and 4 frequent use. No significant correlation was found between these variables and the results of the experiment.

The results of this subjective evaluation present several differences from the previous results and can be seen on Table 4. In this case, the baseline model attained the best result, with an overall quality of experience of 3.91 out of 5.0, followed by the depth filter with a 3.55 score and lastly AsymFormer with a 3.02. As it can be seen from Figure 15, this can be associated with distinct factors. First, the segmentation results advert that users perceived better segmentation quality with Thundernet. Taking into consideration the participant comments, this is due to two factors: (i) the reduced FOV of the RealSense camera compared to the stereo, that prevented the users to see their complete body, thus not appreciating the complete segmentation; (ii) the point cloud representation *per se*. The majority of users complained about seeing their body "segmented". That is, not seeing their body in parts were the depth sensor could not compute depth, which is inherited for this kind of 3D visualization. Secondly, some users found really annoying the delay caused by AsymFormer which is the slowest model and made the users feel the mask was "following them" when moving quickly. Some participants noted a slight (but not annoying) delay with Thundernet, but the best experience was clearly with the depth filter, which functions in real time. Lastly, most participants coincided that the Thundernet experience was better overall mainly because it was a more fluid representation with less delay, flickering and blind spots. Nevertheless, some participants preferred the depth filter experience, noting its capability to interact with a 3D virtual scene. This positions the depth filter as a promising candidate, as it provides depth awareness without delays and blinking false positives.

Despite this, when comparing the amount of false positives, the users assessed that AsymFormer presented the least, which supports previous results. The comparison between Thundernet and the depth filter is surprising: the final score for false positives was similar. Most participants found "blinking" false positives in Thundernet, which were usually considered more annoying than the ones presented in the depth filter, because those were "always there". One participant did not even note the (visible) floor as a false positive in the depth filter. Additionally, both of the point cloud representations got a higher score for distance perception, which proves the original hypothesis. Users with no experience in VR, could not articulate why the distance perception was better with a point cloud, only that their body seemed to have the "correct" proportions, but more experienced users observed that their body was in 3D and noted how this kind of representation might be optimal for interacting with the 3D virtual scene, but preferred to use the depth filter better than AsymFormer if the application supports it, because it has no delay.

Taking all of this into account, it seems that the use of depth does in fact increase the segmentation performance and distance perception, but the point cloud visualization may not be optimal given the current hardware limitations. For interacting with the 3D virtual scene, a depth filter may suffice and to use AsymFormer in a complex environment, the data flow must be optimized. As it can be seen from



(a) Baseline model. (b) Depth-filtered version. (c) AsymFormer model.

Figure 15: Violin plot representation of user scores in the MR-based evaluation for each model. Higher values across all questions indicate better perceived quality.

Table 4, this would imply to optimize all components in the pipeline not just model inference, because, despite the model operating in real-time, the complete pipeline does not satisfy real-time constraints.

8 Discussion

This section will address the research questions presented earlier in the study. For RQ1, the use of depth does improve the perceived quality of the segmentation, which was consistently confirmed through the video-based evaluation, subjective and objectively.

For RQ2, it can be seen that a point cloud does increase the distance perception overall. However, it is important to notice that some novel users could not distinguish between 2D and 3D body representations, so, depending on the final application, it can be considered if a simpler and more plain representation may even increase the overall experience. This is backed by the opinion of the majority of users, which preferred the Thundernet experience in an environment where they only needed to see their body and not interact with anything.

Considering RQ3, there were no specific rigorous questions about embodiment and self-presence in the evaluation, but there are two separate things to consider, the use of depth for segmentation and the use of depth for visualization. The use of depth for segmentation appears to reduce the felt immersion, at least with AsymFormer, due to the delay that makes the user unable to move comfortably at a normal speed in the MR environment, despite not being possible distractions in the background in the form of false positives. For visualization, the results are unclear, on one hand users perceived distances better, which increase the quality of self-perception, but, on the other hand, the limited FOV of the RealSense camera joined with the imperfections inherited by a point cloud conditioned the participants to prefer a 2D representation.

Model	**EgoBodies dataset**		**Video evaluation**		**Unity evaluation**	
	Test IoU	Speed (s)	IoU	MOS	MOS (Overall)	MOS (Distance)
Thundernet Mono	0.895	0.0199	0.72	3.46	–	–
Thundernet Stereo	0.814	0.0199	0.68	3.03	**3.91**	4.25
Depth Filter	–	0.0000	0.32	2.52	3.55	**4.73**
AsymFormer (RGB-D)	**0.926**	0.0329	**0.82**	**4.23**	3.02	4.60

Table 5: Summary of performance across the EgoBodies dataset, video-based evaluation, and MR-based evaluation (overall and distance perception).

Table 5 summarizes the performance accross different evaluation stages. It can be seen that AsymFormer clearly provides the most accurate segmentation results and that a point cloud representation increases

the distance perception. However, for a complex pipeline, the point cloud could only be a suitable option when working with a simple application where the false positives that are not filtered with the depth filter do not affect the MR experience. In other words, a depth filter can effectively complement semantic segmentation results (by discarding false positives in the background), but can not serve as a perfect substitute for it, at least in environments where the user would only want to see a set of concrete objects in a closed environment. In these situations, a depth filter would always include visible nearby objects (for example, the floor) which can reduce the felt immersion. In contrast, in experiences where seeing the floor does not matter, a depth filter can be a better option than semantic segmentation because it works in real-time and it does not involve the process of retraining a ML model with an adapted database (to segment only a set of specific objects).

Future work could explore further sensors that have a wider FOV and a better depth estimation (maybe with an stereo depth sensor to avoid blind spots) or alternative representations with fewer drawbacks. A possible research line could be to replicate this work using the RGB-D incorporated sensors of the Meta Quest 3, which have only recently become accessible (since January). This would provide a wider FOV and better resolution than the RealSense camera.

9 Conclusions

This work has extended previous MR approaches based solely on RGB input by incorporating depth information to enhance user immersion. We have proposed the RGB-D Egocentric Bodies Dataset, which extends the Egocentric Bodies Dataset, by adding the corresponding depth frames. This dataset was used to train the AsymFormer model (RGB-D) and the monocular version of Thundernet. We have designed a pipeline that enables to process RGB-D (or RGB) frames to render the user's body in a 3D (or 2D) visualization within a Unity environment, by segmenting the users body using different segmentation models. Additionally, we have conducted two sets of evaluations to determine the effect of depth on segmentation accuracy and distance perception. The experiments were conducted separately to isolate the specific impact of depth on each dimension.

This project demonstrates that depth-enhanced egocentric segmentation significantly improves both segmentation accuracy and the immersive experience in MR applications. Incorporating depth has been shown to effectively refine the quality of the segmentation, as seen with AsymFormer, and the 3D point cloud visualization improved depth perception, that could be especially useful in scenarios involving spatial interaction. However, the limitations of current depth sensors, such as missing or unstable points and lack of definition, also affect the quality of the 3D reconstruction. Additionally, the high computational cost of models like AsymFormer may not justify their use in all contexts. For non-interactive scenarios, simpler 2D segmentation may suffice, as they offer a more user-friendly alternative without substantially decreasing the overall experience.

Future work could explore alternative 3D representations that are more visually stable, or real-time optimizations for using more complex models that involve optimizing all components in the pipeline, not just inference time. Improvements in technology, such as depth sensors with a wider FOV, could further enhance the utility of depth-aware segmentation in MR environments by providing an even more realistic self-avatar.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *The 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019.

[2] Ferran Argelaguet, Ludovic Hoyet, Michaël Trico, and Adrien Lecuyer. The role of interaction in virtual embodiment: Effects of the virtual hand representation. In *2016 IEEE Virtual Reality (VR)*,

pages 3–10, 2016.

[3] Gerd Bruder, Frank Steinicke, Kai Rothaus, and Klaus Hinrichs. Enhancing presence in head-mounted display environments by visual body feedback using head-mounted cameras. In *2009 International Conference on CyberWorlds*, pages 43–50, 2009. doi: 10.1109/CW.2009.39.

[4] Siqi Du, Weixi Wang, Renzhong Guo, Ruisheng Wang, and Shengjun Tang. Asymformer: Asymmetrical cross-modal representation learning for mobile platform real-time rgb-d semantic segmentation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 7608–7615, 2024. doi: 10.1109/CVPRW63382.2024.00756.

[5] ELP. Elp 960p hd ov9750 high frame rate mjpeg 60fps uvc otg stereo webcam dual lens, 2025. URL https://www.webcamerausb.com/elp-960p-hd-ov9750-high-frame-rate-mjpeg-60fps-uvc-otg-stereo-webcam-dual-lens-p-159.html.

[6] Ester Gonzalez-Sosa, Pablo Pérez, Ruben Tolosana, Redouane Kachach, and Alvaro Villegas. Enhanced self-perception in mixed reality: Egocentric arm segmentation and database with automatic labeling. *IEEE Access*, 8:146887–146900, 2020. doi: 10.1109/ACCESS.2020.3013016.

[7] Ester González-Sosa, Guillermo Robledo, Diego González Morín, Pablo Perez-Garcia, and Álvaro Villegas. Real time egocentric object segmentation: Thu-read labeling and benchmarking results. *ArXiv*, abs/2106.04957, 2021. URL https://api.semanticscholar.org/CorpusID:235377314.

[8] Ester González-Sosa, Andrija Gajic, Diego González Morín, Guillermo Robledo, Pablo Pérez, and Álvaro Villegas. Real time egocentric segmentation for video-self avatar in mixed reality. *ArXiv*, abs/2207.01296, 2022. URL https://api.semanticscholar.org/CorpusID:250264412.

[9] Yaosi Hu, Zhenzhong Chen, and Weiyao Lin. Rgb-d semantic segmentation: A review. In *2018 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–6, 2018.

[10] Intel Corporation. Intel® realsense™ depth camera d435, 2025. URL https://www.intelrealsense.com/depth-camera-d435/.

[11] Intel Corporation. Intel realsense sdk 2.0 (librealsense). https://github.com/IntelRealSense/librealsense, 2025. Accessed: 2025-05-18.

[12] Christian Keimel, Julian Habigt, Clemens Horch, and Klaus Diepold. Qualitycrowd - a framework for crowd-based quality evaluation. In *Picture Coding Symposium 2012 (PCS2012)*, pages 245–248, May 2012. doi: 10.1109/PCS.2012.6213338.

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[15] Gun Lee, Joshua Chen, Mark Billinghurst, and Robert Lindeman. Enhancing immersive cinematic experience with augmented virtuality. pages 115–116, 09 2016. doi: 10.1109/ISMAR-Adjunct.2016.0054.

[16] Kwan Min Lee. Presence, explicated. *Communication Theory*, 14(1):27–50, 2004. doi: 10.1111/j.1468-2885.2004.tb00302.x. URL https://academic.oup.com/ct/article/14/1/27/4110793.

[17] Xiang Li, Heqian Qiu, Lanxiao Wang, Hanwen Zhang, Chenghao Qi, Linfeng Han, Huiyu Xiong, and Hongliang Li. Challenges and trends in egocentric vision: A survey, 2025. URL https://arxiv.org/abs/2503.15275.

[18] Mohammad Moghimi, Pablo Azagra, Luis Montesano, Ana C. Murillo, and Serge Belongie. Experiments on an rgb-d wearable vision system for egocentric activity recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 611–617, 2014. doi: 10.1109/CVPRW.2014.94.

[19] Diego Gonzalez Morin, Ester Gonzalez-Sosa, Pablo Perez, and Alvaro Villegas. Full body video-based self-avatars for mixed reality: from e2e system to user study. *Virtual Reality*, 27(3):2129–2147, 2023. URL https://doi.org/10.1007/s10055-023-00785-0.

[20] Zheng Qin, Zeming Li, Zhaoning Zhang, Yiping Bao, Gang Yu, Yuxing Peng, and Jian Sun. Thundernet: Towards real-time generic object detection on mobile devices. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6717–6726, 2019. URL https://api.semanticscholar.org/CorpusID:208005134.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[22] Muhammad Shaikh and Douglas Chai. Rgb-d data-based action recognition: A review. *Sensors*, 21: 4246, 06 2021. doi: 10.3390/s21124246.

[23] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 2017. doi: 10.1109/TPAMI.2016.2572683.

[24] Richard Skarbez, Missie Smith, and Mary C Whitton. Revisiting milgram and kishino's reality-virtuality continuum. *Frontiers in Virtual Reality*, 2:647997, 2021.

[25] Yansong Tang, Yi Tian, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Action recognition in rgb-d egocentric videos. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3410–3414, 2017. doi: 10.1109/ICIP.2017.8296915.

[26] Yansong Tang, Zian Wang, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Multi-stream deep neural networks for rgb-d egocentric action recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(10):3001–3015, 2019. doi: 10.1109/TCSVT.2018.2875441.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[28] T. Waltemate, D. Gall, D. Roth, M. Botsch, and M. E. Latoschik. The impact of avatar personalization and immersion on virtual body ownership, presence, and emotional response. *IEEE Transactions on Visualization and Computer Graphics*, 24(4):1643–1652, 2018. doi: 10.1109/TVCG.2018.2794629. URL https://ieeexplore.ieee.org/document/8263407/.

[29] Wei Xiang, Hongda Mao, and Vassilis Athitsos. Thundernet: A turbo unified network for real-time semantic segmentation. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1789–1796, 2019. doi: 10.1109/WACV.2019.00195.

[30] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[31] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[32] Zhe Yang, Baozhong Mu, Mingxun Wang, Xin Wang, Jie Xu, Baolu Yang, Cheng Yang, Hong Li, and Rongqi Lv. Efinet: Efficient feature interaction network for real-time rgb-d semantic segmentation. *IEEE Access*, 12:151046–151062, 2024. doi: 10.1109/ACCESS.2024.3478746.

Appendices

A Step by step compute for LAFS and CMA modules

The LAFS and CMA modules from AsymFormer are now described in detail. LAFS is the module that selects key features from both branches while CMA mixes the features from both branches together. Below there is an step by step compute of the output from LAFS. Figure 3 can serve as a visual guide to follow these steps.

1) The input of the LAFS module is a concatenation in the channel dimension of the depth and color features:
   - RGB Features: $F_{RGB} \in \mathbb{R}^{C_{RGB} \times H \times W}$
   - Depth Features $F_{Depth} \in \mathbb{R}^{C_{Depth} \times H \times W}$
   - => Input = $F_{in} \in \mathbb{R}^{C \times H \times W}$ where $C = C_{RGB} + C_{Depth}$
2) It extracts global information using an adaptive average pooling:

$$Avg = \text{Adaptive Average Pooling}(F_{in}) \in \mathbb{R}^{C \times 1 \times 1} \tag{1}$$

3) Now, the LAFS module has two parts, one to extract channel attention and one to extract spatial attention. At the same time, both attention weights are computed.

   3.1) **Channel Attention Weights:** The Avg tensor passes through a Squeeze-Excitation (SE) network and a sigmoid to normalize the weights. This will result in the attention weights for the channels.

$$W_C = \sigma(\text{MLP}(Avg)) \in \mathbb{R}^{C \times 1 \times 1} \tag{2}$$

   3.2) **Spatial Attention Weights:** The Avg tensor passes through a SE network, obtaining the R_{avg} vector.

$$R_{avg} = (\text{MLP}(Avg)) \in \mathbb{R}^{C \times 1 \times 1} \tag{3}$$

   Now, we compute the similarity between R_{avg} and F_{in}

$$I_s = \text{Dot}(F_{input}^T, Avg) \in \mathbb{R}^{1 \times H \times W} \tag{4}$$

   With that, the attention spatial weigths can be obtained.

$$W_S = \sigma\left(\frac{I_s}{C^2}\right) \in \mathbb{R}^{1 \times H \times W} \tag{5}$$

4) With this weights, the output can be computed:

$$F_{LAFS} = F_{input} \cdot W_C \cdot W_S \in \mathbb{R}^{C \times H \times W} \tag{6}$$

Now, we will compute the CMA module step by step, that is, we will explain how to compute the similarity between the branches (cross-modal self-similarity). Figure 4 provides a visual representation of the steps.

1. The input of the module will be the RGB features, the depth features and the fused features (LAFS output)
   - RGB Features: $F_{RGB} \in \mathbb{R}^{C_{RGB} \times H \times W}$
   - Depth Features: $F_{Depth} \in \mathbb{R}^{C_{Depth} \times H \times W}$
   - Fused Features: $F_{LAFS} \in \mathbb{R}^{C \times H \times W}$
2. Now, we will obtain the key, query and value matrices.

2.1) The Key and Query tensors will come from the RGB and depth features. The RGB features will produce a key and a value tensor, and the depth features another.
- $K_{RGB} = key(F_{RGB}) \in \mathbb{R}^{C_{K_{RGB}} \times H \times W}$
- $K_{Depth} = key(F_{Depth}) \in \mathbb{R}^{C_{K_{Depth}} \times H \times W}$
- $Q_{RGB} = query(F_{RGB}) \in \mathbb{R}^{C_{Q_{RGB}} \times H \times W}$
- $Q_{Depth} = query(F_{Depth}) \in \mathbb{R}^{C_{Q_{Depth}} \times H \times W}$

2.2) The value tensor will come from F_{LAFS}. We will obtain a matrix which will be then split in two matrices: V_1 and V_2

$V_1, V_2 = split(value(F_{LAFS})) \in \mathbb{R}^{C_V \times H \times W}$
Where the key, query and value operations model the usual procedure to obtain key, queries and values (with trainable weights matrices).

3. Now, we will combine the information from the depth and RGB branches. For that, we will obtain a Key and Query matrices by concatenating the previous matrices. Then, to ensure combination, this tensors will be shuffled in the channel dimensions.
   - $K = shuffle(cat(K_{RGB}, K_{Depth})) \in \mathbb{R}^{(C_{K_{RGB}} + C_{K_{Depth}}) \times H \times W}$
   - $Q = shuffle(cat(Q_{RGB}, Q_{Depth})) \in \mathbb{R}^{(C_{Q_{RGB}} + C_{Q_{Depth}}) \times H \times W}$
4. We divide these tensors into K_1, Q_1, K_1, Q_2 and perform dot product attention with the value tensors:

$$W_1 = Softmax\left(\frac{Q_1 \cdot K_1^T}{\sqrt{C_1}}\right) \quad \text{and} \quad W_2 = Softmax\left(\frac{Q_2 \cdot K_2^T}{\sqrt{C_2}}\right)$$

   Where C_1 and C_2 represent the dimensions of the query vectors Q_1 and Q_2 respectively.
5. Finally, the output from the CMA module is obtained by combining the weighted value tensors:

$$F_{CMA} = cat(W_1 \cdot V_1, W_2 \cdot V_2) \in \mathbb{R}^{C \times H \times W}$$

Finally, the output from this modules (LAFS + CMA) will be the sum of F_{CMA} and F_{LAFS}. This will serve as the input of the depth intermidiate modules. At the end, the information from the last LAFS + CMA module is passed through an MLP network to output the final segmentation result.

B Training and validation graphs

B.1 Thundernet Mono

From figures 16 and 17, it can be seen that Thundernet does not present any signs of overfitting or vanishing gradients.

B.2 AsymFormer

From figures 20 and 21, it can be seen that AsymFormer does not present any signs of overfitting or vanishing gradients. It is also noticeable the differences between figures 18 and 19 and figures 20 and 21, as the training and validation curves seem smoother for the version on AsymFormer trained solely on synthetic images. However, as shown in the results, these did not imply a better performance, as the use of real depth maps provide an increase in segmentation accuracy.

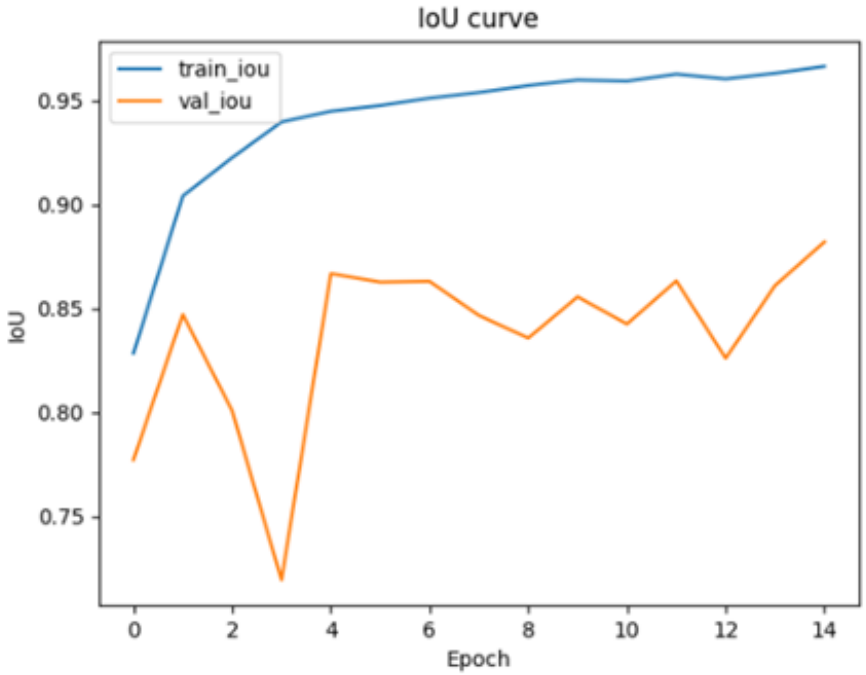


Figure 16: IoU evolution during the training of the monocular version of Thundernet.

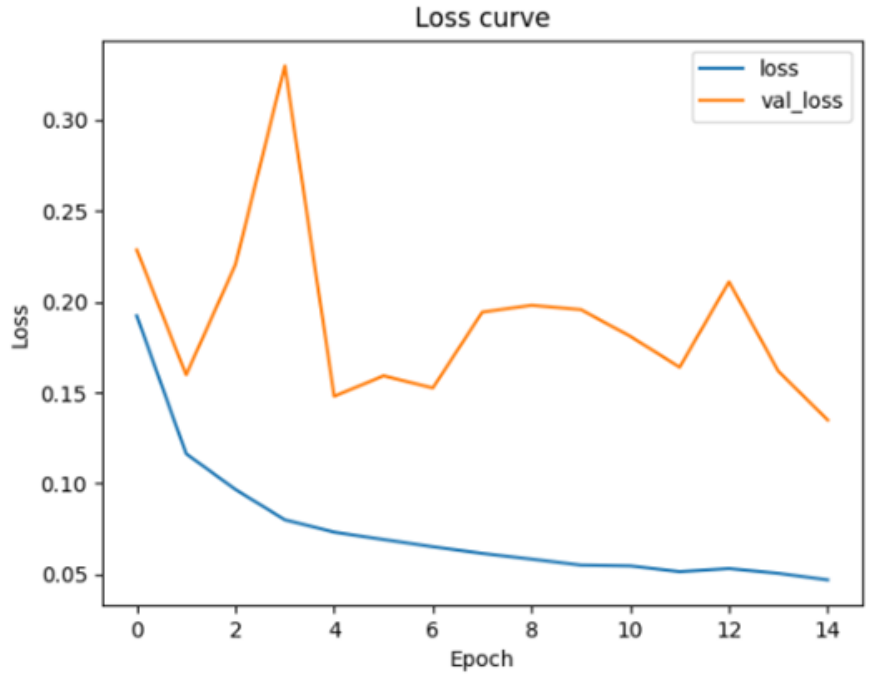


Figure 17: Loss evolution during the training of the monocular version of Thundernet.

Figure 18: Training evolution curves for the AsymFormer model. From left to right, the curves represent the evolution of accuracy, mIoU and loss.

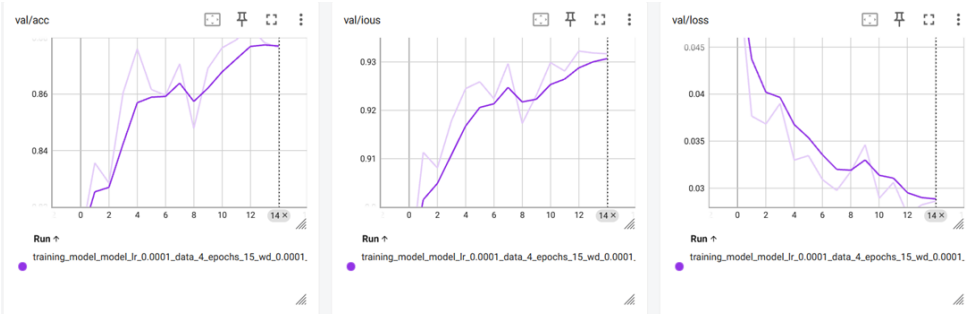


Figure 19: Validation evolution curves for the AsymFormer model. From left to right, the curves represent the evolution of accuracy, mIoU and loss.

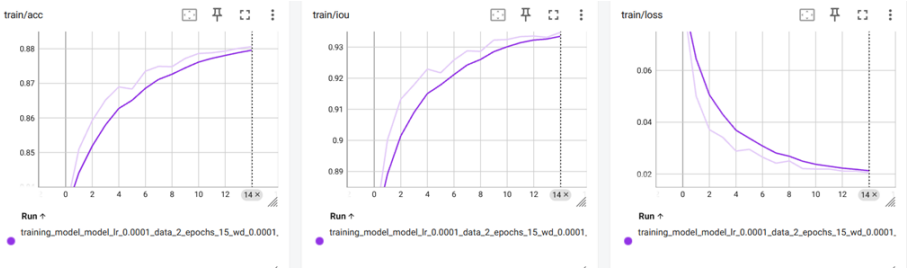


Figure 20: Training evolution curves for the AsymFormer model trained solely on synthetic depth data. From left to right, the curves represent the evolution of accuracy, mIoU and loss.

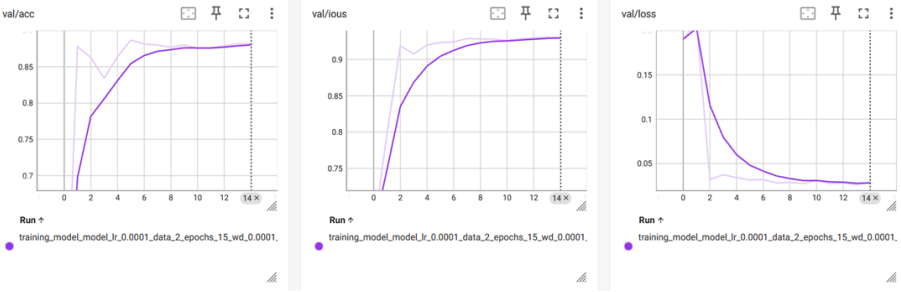


Figure 21: Validation evolution curves for the AsymFormer model trained solely on synthetic depth data. From left to right, the curves represent the evolution of accuracy, mIoU and loss.