Temporally-Aware Deep Learning for Egocentric Indoor Scene Simplification in Bionic Vision

Yanxiu Jin*
Department of Electrical and Computer Engineering
University of California, Santa Barbara
Santa Barbara, CA, United States
yanxiu_jin@ucsb.edu

Ron Kibel*
Department of Computer Science
University of California, Santa Barbara
Santa Barbara, CA, United States
rkibel@ucsb.edu

***Abstract*—Retinal degeneration is one of the leading causes of incurable low vision worldwide. For many visually impaired individuals, advancements in retinal prostheses offer a potential means of restoring limited visual perception by electrically stimulating neurons in the retina to generate artificial percepts, commonly known as phosphenes. Nevertheless, although these phosphenes convey relevant visual information, stimulation via an electrode array in the eye is still a rather limited technology for producing high-resolution representations of the real world. Instead, recent studies have explored a deep learning (DL) approach for scene simplification via an external visual processing unit (VPU), which sends the processed video to the implant downstream. Using a fusion of saliency, image segmentation, and depth estimation, models show promise in enhancing prosthetic vision through scene simplification, but often suffer from artifacts, flickers, and fragmentation, especially across video frames. Moreover, scene simplification strategies in complex, cluttered indoor spaces remain largely unexplored. In this work, we present an improved scene simplification pipeline that integrates refined object segmentation with saliency and depth estimation, specifically targeted to improve temporal consistency in *indoor egocentric environments*. A mixture of qualitative and quantitative results indicate improvement on indoor depth and segmentation baselines, with greater consistency across frames in scene and object simplification. The code for our paper is available at https://github.com/rkibel/BionicVision.**

***Index Terms*—retinal prosthesis, scene simplification, deep learning, image segmentation, temporal consistency**

I. Introduction

Retinal degenerative diseases, such as age-macular degeneration (AMD) and retinitis pigmentosa (RP) are significant contributors to irreversible vision loss worldwide. AMD alone affects approximately 200 million people globally [7]. These conditions contribute to the loss of photoreceptors in the retina, leading to a degradation of visual acuity and ultimately, blindness. In some cases, however, individuals can be fitted with a visual neuro-prosthesis (a bionic eye) that functions by electrically stimulating clusters of remaining retinal neurons to evoke neuronal responses, interpreted by the brain as percepts known as phosphenes. These phosphenes help patients distinguish between high-contrast representations and provide basic orientation and navigation signals.

* Both authors contributed equally to this research.

However, the success of modern retinal implants in providing detailed visual signals is impeded by a few factors. For one, underlying physiology has shown significant phosphene distortion between subjects, severely altering the anticipated quality of retinal electrode arrays [2]. Moreover, factors like limited electrode count, large electrode size, neuro-electronic signal distortions, and small fields of view indicate that prosthetic vision is far from restoring "natural" vision [1].

These limitations motivate the search to maximize the amount of useful information conveyed by a visual prosthetic system rather than just to attempt to restore vision. One such approach is through image processing techniques to simplify and enhance raw video before it is fed into the implant, a process known as **scene simplification**. Deep-learning-based scene simplification techniques have emerged as a powerful way to extract and simplify important visual data, but existing methods invoke static (per-frame) analysis, and as a result tend to produce artifacts, flickers, and fragmented representations that do not align across frames. This issue is particularly pronounced in indoor environments, where clutter and rapid object movements relative to the field of view present significant challenges.

To address these challenges, we propose a pipeline that emphasizes temporal consistency, tested on indoor datasets. Our approach focuses on three primary objectives:

1) Enhance temporal consistency in existing algorithms.
2) Integrate state-of-the-art models to improve scene simplification using a fusion of temporally consistent saliency, segmentation, and depth.
3) Evaluate modifications with both qualitative and quantitative egocentric (first-person) benchmarks.

II. Related Work

Scene simplification via deep learning has been pivotal in extracting important information for visual prostheses, specifically for tasks in navigation. For example, Hicks et al. (2013) developed a head-mounted visual display system with depth extraction to highlight objects based on size and distance, and demonstrated rapid skill acquisition when using this kind of display for obstacle avoidance [12]. More recently, Rasla and Beyeler (2022) explored the relative importance of depth and semantic cues for indoor mobility using simulated

prosthetic vision (SPV) in virtual reality, and concluded that an ability to switch between depth and semantic modalities can significantly improve navigation [23]. In general, flexible scene simplification pipelines are essential both to increase the learning rate of users and to enhance task performance.

In recent years, researchers have also developed increasingly sophisticated scene simplification pipelines. For instance, Han et al. (2021) [11] focuses on user classification of cars and people with a computational model of the retina to generate realistic SPV predictions. Their scene simplification pipeline integrates three primary modalities:

- Saliency mapping with *DeepGaze II* [17] to highlight regions of an image that are likely to attract human attention
- A combination of object segmentation with *detectron2* [24] and scene segmentation with *MIT Scene Parsing Benchmark* [26] to identify and segment objects based on semantic understanding
- Monocular depth estimation with *Monodepth2* [9] to estimate object distance

We will delve further into the importance of these models in the following section. Ultimately, Han et al. concluded that segmentation was the most relevant modality given their use case (an outdoor environment), suggesting that visual saliency and depth estimation had minimal impact as auxiliary components. However, when adapting these methods to indoor contexts, it is crucial to reassess the relevance of each component—a goal we aim to accomplish in this work.

III. Methods

Fig. 1. RGB frame of our evaluation video, processed in 20 fps (frame 11)

A. Baseline

We treat the work by Han et al. [11] in the previous section as our baseline. Specifically, we aim to improve on this pipeline for indoor uses, targeting temporal consistency to reduce object fragmentation.

First, we delve into the four different scene simplification strategies proposed by the paper:

1) Saliency-based: DeepGaze II [17] is a static (per-frame) saliency model that takes an image as input and outputs a continuous heatmap, where regions most likely to attract human attention are given higher intensities and less relevant background details are discarded or assigned lower intensities. Since it solely relies on CNN-derived features to analyze low-to-mid-level visual cues such as color contrasts, edges, and textures, the emphasized regions in the output implicitly correspond to semantically meaningful areas—humans are naturally drawn to such visual cues. These areas often align with critical elements necessary for recognition, interaction, and scene understanding. The baseline linearly maps importance values from the saliency heat map to stimulus amplitudes in the simulated retinal implant.

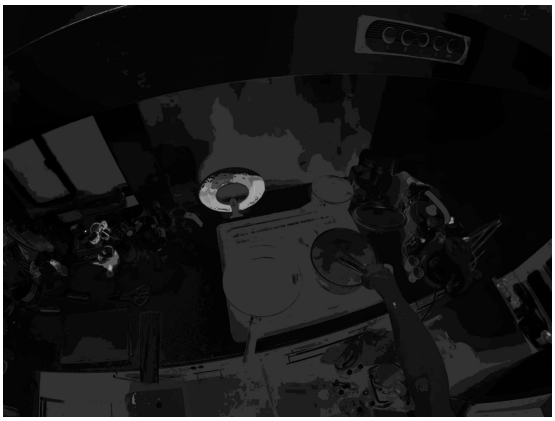

Fig. 2. Saliency map from DeepGaze 2 (frame 11)

2) Segmentation-based: The baseline uses an image-based scene parsing algorithm from the MIT Scene Parsing Benchmark [26] to obtain scene segmentation masks with predefined labels. Following this parsing, a series of cleanup stages are performed:

1) Small objects in the mask are classified as artifacts and removed.
2) An edge detection filter is applied to the cleaned mask, followed by dilation.
3) The Probabilistic Hough Transform is applied to extract prominent straight lines from the edge map that satisfy minimum line length and maximum line gap criteria. Among the lines, only those with both horizontal and vertical differences between endpoints exceeding 5 pixels are retained. This filters out noise and performs, but **only excludes very short segments, thus it may not fully cover all edges near image boundaries.** The result is then dilated for better visibility
4) For the first 10 frames, the edge maps are stored in a history buffer. Starting from the 11th frame, the baseline conducts a temporal max pooling of size 10. However, **the baseline history buffer does not update—it counterintuitively only maintains the 1st-10th frames as history.**

5) The result is refined with a Probabilistic Hough Transform, followed by erosion and binary thresholding. However, **the thresholding step is redundant as dilate and erosion do not create-non binary values**.
6) A morphological opening is applied, followed by a second Hough Transform and erosion.

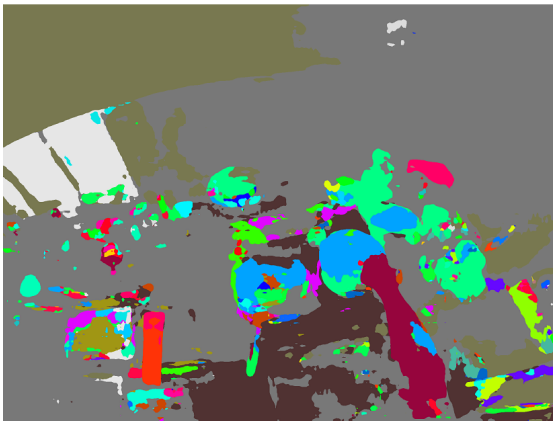

Fig. 3. MIT scene segmentation result (frame 11)

For object segmentation, the baseline uses detectron2 [24], an image-based object segmentation algorithm to obtain masks of important objects with pre-defined labels. If no important objects are detected, scenes edges are displayed. Otherwise, scene edges aren't displayed. The resulting binary masks are then linearly mapped to stimulus amplitudes in the simulated retinal implant.

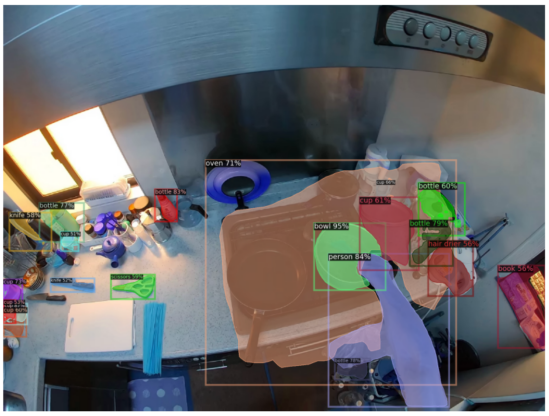

Fig. 4. Detectron2 result (frame 11)

3) Depth-based: The baseline uses Monodepth2[9], specifically mono+stereo_640x192 to estimate a depth map from an input image. It removes the farthest 20% of pixel depths (preserves the 80th percentile), and applies an exponential decay, mapping the closest pixels to grayscale 180 and the farthest pixels to 0. Thus the baseline enhances depth perception by emphasizing nearer objects.

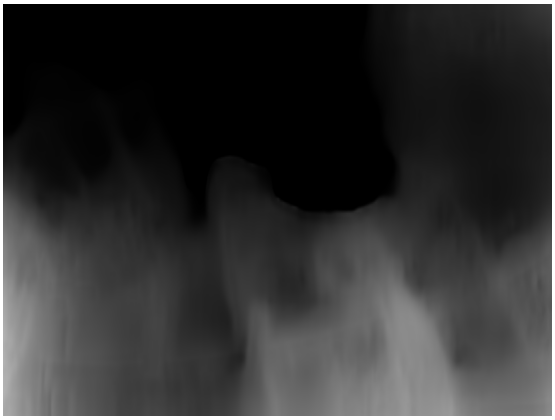

Fig. 5. Clipped depth map from Monodepth2 (frame 11)

4) Combination: The baseline thresholds the saliency map to retain only the 10 % most salient pixels and combines it with the segmentation map using a logical OR. The grayscale value of each pixel is then scaled quadratically with depth. In this way, the static saliency can help highlight the regions of interest where segmentation models fail. Moreover, nearby obstacles can be highlighted by the depth algorithm.

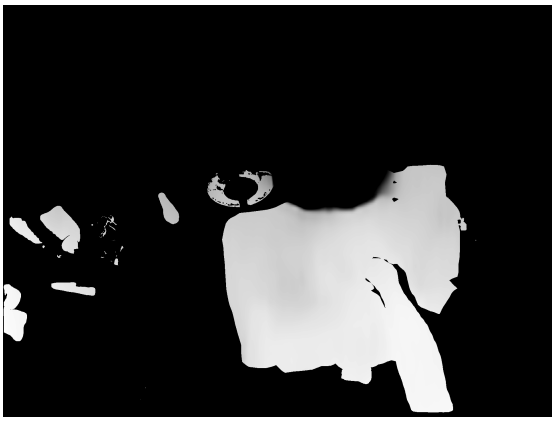

Fig. 6. Combination: 50% saliency, clipped, quadratic (frame 11)

B. Segmentation-based Improvements

We first refine the existing segmentation pipeline for indoor, temporal contexts.

1) Baseline Improvements: First, we eliminate artifacts on scene by using connected component analysis (area threshold of 1500) before the first Probabilistic Hough Transform and after the final dilation (erosion applied before second elimination).

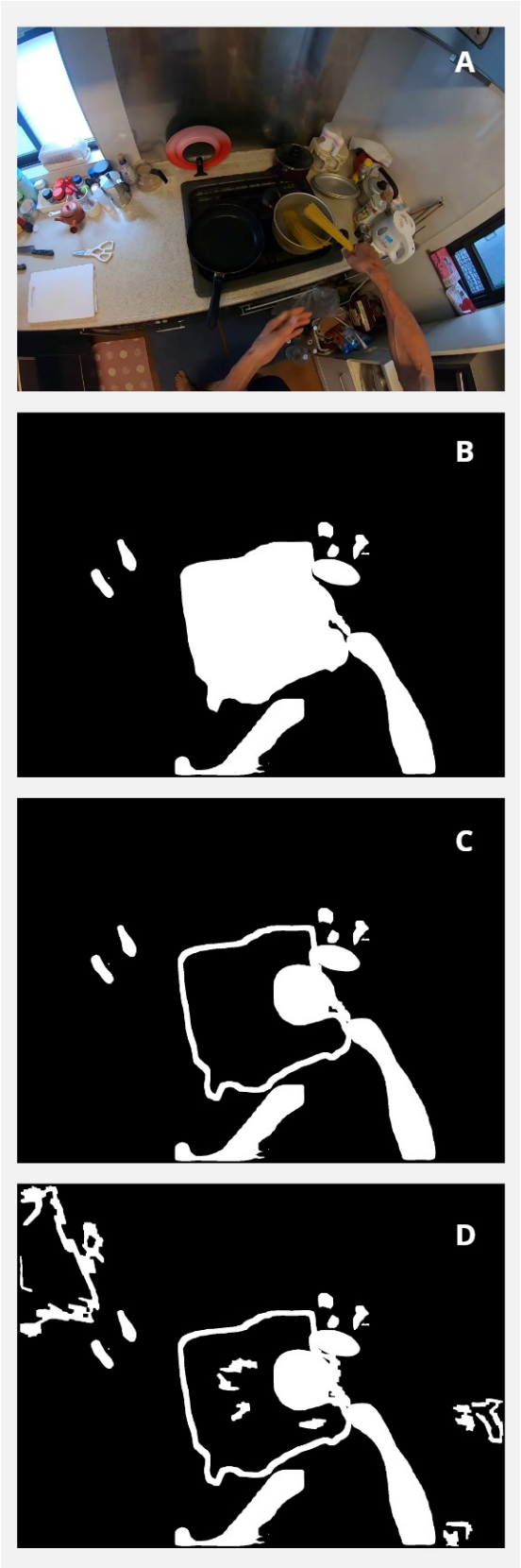

Fig. 7. Detectron2 modifications (frame 157) to merge scene and object segmentations. A) pure RGB frame. B) baseline important object segmentation. C) edge retrieval of large objects. D) scene edges re-added.

Next, we treat large objects as scenes, hence converting them to edges to eliminate occlusion in important objects. We decide to always display scene segmentation along with important objects, as indoor scenes like windows and doors provide necessary cues for orientation indoors. The visual results for sample frame 157 are shown in Fig. 7.

Lastly, we have a few extra suggestions for baseline segmentation. Given the limitations discussed in the previous section (in bold), we now check if edges are sufficiently long (over 30 pixels) and not located too close to the image boundaries (within a 10-pixel margin). We also fix the buffer history by updating with past 10 frames and remove the binary thresholding step. We configure the pipeline with optimal parameters suitable for our Ego4D dataset (minimum line length, kernel sizes).

2) Temporal Segmentation Model: We introduce temporal consistency in object segmentation using DEVA [4], a decoupled video-based segmentation approach for 'tracking anything'. Combined with SAM (Segment Anything Model) [15], DEVA demonstrates state-of-the-art performance in open-world large-vocabulary video segmentation. It also allows prompting for segmentation, which provides great potential for future adaptive segmentation (user-specified real-time masks).

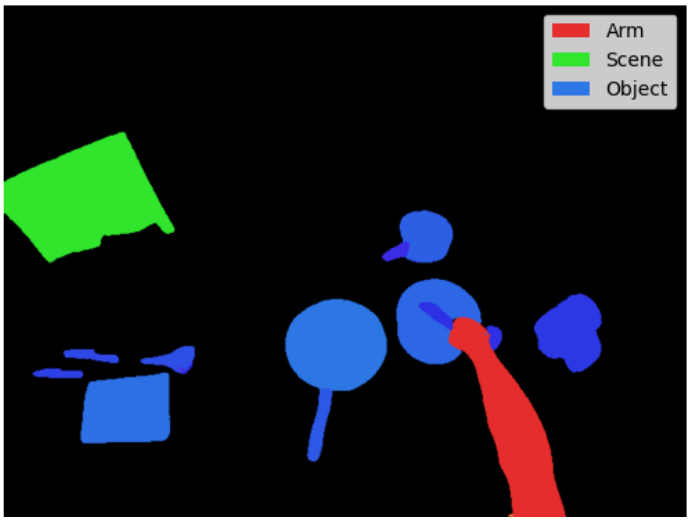

Fig. 8. Colored DEVA outputs of arm, scene, and object instances (frame 11). These results were extracted using natural language prompts, automatically classified, and masked.

3) Assigning Brightness on Priority: We propose a priority table in Table 1 to determine which features should be emphasized and correspondingly assigned a higher brightness.

Priority Level	Elements	When (Brightness)	Reasoning
High	Hands / Arms	Always (b1)	Cognition Feedback
Mid-High	Scene Objects	No objects detected (b1) Objects detected (b3)	Navigation, Orientation
Mid-High	Primary Objects	Near hand (b2) Near Gaze (b1)	Interactions
Mid	Secondary Objects	No arms, scene (b2) Arms, scene (b3)	Situational awareness
Low	Clutter / Artifacts	Never, removed	Reduce clutter

TABLE I
PRIORITY TABLE (USING BRIGHTNESS AS INDICATION OF PRIORITY, B1=255; B2=220; B3=160)

Interviewing a retinal implant patient, we determined that visual feedback view body position can prove essential to improving task effectiveness—as a result, we assign arms the highest priority and brightness (b1). Moreover, we give medium-high priority to scene objects (b1) and choose to always display scene edges, for similar reasons in improving orientation. Note that when objects are detected, the scene brightness drops (b3).

One of the ways we determine primary objects is through proximity to detected hands (within a circle around a hand). The radius of the circle follows an exponential decay function based on the distance from the hand position to the image center. The Intuition here is that the closer to the center, the more likely the user is focusing on this area, so a larger "actionable" radius is given. More towards the edges, the hand might have just entered the frame or is not yet ready for interaction, so the radius can be reduced.

First, we compute the maximum possible distance from the image center to any corner:

$$d_{\max} = \sqrt{\left(\frac{w}{2}\right)^2 + \left(\frac{h}{2}\right)^2}.$$

where w and h are the width and height of the image, respectively.

Next, we define d, the Euclidean distance between the hand position $(x_{\text{hand}}, y_{\text{hand}})$ and the image center (c_x, c_y). Using d, we compute the squared exponential decay ratio c:

$$c = e^{-\alpha\left(\frac{d}{d_{\max}}\right)^2}$$

where α is a decay parameter, a larger value means the radius decreases more rapidly with distance. Note that squaring makes the exponential decay much slower near the center and much faster towards the edges. To ensure the ratio remains within valid bounds, we clip it between 0 and 1, and dynamically adjust the radius:

$$r = r_{\min} + (r_{\max} - r_{\min}) \cdot c$$

where $r_{\min}$ and $r_{\max}$ are predefined smallest radius and largest radius respectively. In our implementation, $r_{\min}$ is 100, $r_{\max}$ is 400 and α is 2.0. This produces the squared exponential map in Figure 9.

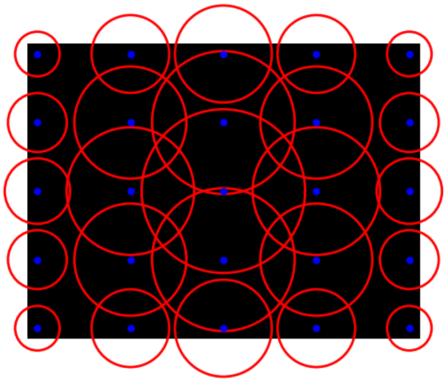

Fig. 9. Squared exponential circle map to determine primary objects close to hand, $\alpha = 2.0$

In order to estimate the hand's coordinate, we need to know the orientation of the arm. Orientation is estimated using Principal Component Analysis (PCA). Given a binary mask where the arm pixels are represented by white pixels (255), we extract the coordinate set x_s, y_s:

$$(x_s, y_s) = \{(x_i, y_i) \mid \text{mask\_image}(x_i, y_i) = 255\}$$

where (x_s, y_s) represents all object pixel locations.
The centroid of the object is calculated as:

$$\bar{x}_s = \frac{1}{N}\sum_{i=1}^{N} x_i, \quad \bar{y}_s = \frac{1}{N}\sum_{i=1}^{N} y_i$$

where N is the total number of object pixels and x_i, y_i denote the detected locations of the arm. We then center each arm-detected location (x_s, y_s) by subtracting the centroid

$$x'_s = x_s - \bar{x}_s, \quad y'_s = y_s - \bar{y}_s$$

and compute the 2×2 Covariance Matrix:

$$\mathbf{C} = \begin{bmatrix} \text{Var}(x'_s) & \text{Cov}(x'_s, y'_s) \\ \text{Cov}(x'_s, y'_s) & \text{Var}(y'_s) \end{bmatrix},$$

where

$$\text{Var}(x'_s) = \frac{1}{N}\sum_{i=1}^{N}(x'_i - \bar{x}_s)^2, \quad \text{Var}(y'_s) = \frac{1}{N}\sum_{i=1}^{N}(y'_i - \bar{y}_s)^2,$$

$$\text{Cov}(x'_s, y'_s) = \frac{1}{N}\sum_{i=1}^{N}(x'_i - \bar{x}_s)(y'_i - \bar{y}_s).$$

Now we can solve for eigenvalues λ and eigenvectors v and can determine the Principal Axis, the eigenvector corresponding to the largest eigenvalue. Eventually, we can compute the orientation angle θ of the principal axis relative to the negative y-axis as

$$\theta_{\text{deg}} = \arctan 2(x_s, y_s) \cdot \frac{180}{\pi}$$

where (x_s, y_s) are the components of the principal eigenvector v_{p}. The computed angle is measured counterclockwise from the negative y-axis.

We approximate the hand's coordinate using the bounding box corners and θ:

$$y_{\min} = \min(y_s), \quad y_{\max} = \max(y_s)$$
$$x_{\min} = \min(x_s), \quad x_{\max} = \max(x_s)$$
$$\theta_{\text{deg}} < 0 \implies (c_x, c_y) = (x_{\min}, y_{\min})$$
$$\theta_{\text{deg}} > 0 \implies (c_x, c_y) = (x_{\max}, y_{\min})$$
$$\theta_{\text{deg}} = 0 \implies (c_x, c_y) = \left(\frac{x_{\min} + x_{\max}}{2}, y_{\min}\right)$$

where (c_x, c_y) represents the approximated hand centroids. Above equations are based on NumPy coordinate system, where the top-left corner is (0,0), x increases to the right, and y increases downward.

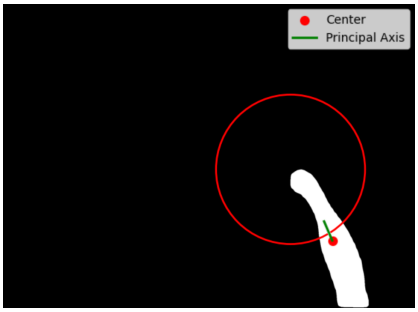

Fig. 10. PCA Principal Axis of an arm and circled hand location. Objects within the circle will be classified as primary objects.

Each object instance is considered "near-at-hand" if more than 50% of the pixels lie within the circle (brightness b2). Otherwise, it will be assigned b3. A history of the past five frames and most recent hand circles is maintained to ensure that if an arm is temporarily undetected due to segmentation errors, the algorithm can refer to recent frames to infer continuity.

For secondary objects, if there are arms detected in the scene, they will have a brightness b3. If arms have not appeared for 30 frames, secondary objects will have brighness b2. All the contents except **Near Gaze** in the table are implemented by now.

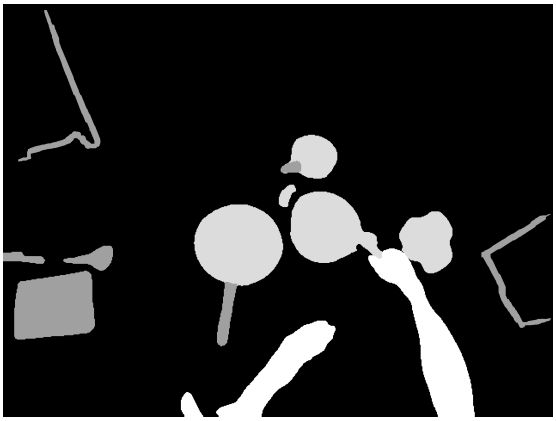

Fig. 11. Priority table(No gaze) on DEVA segmentation (frame 157)

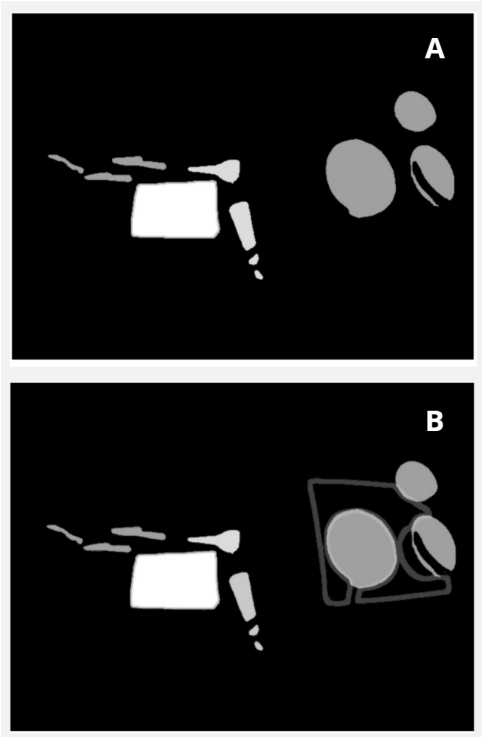

Fig. 12. Priority table(No gaze) on DEVA segmentation (frame 97). A) before persistence and averaging. B) after persistence and averaging.

4) Persistence and Averaging Across Frames: We introduced two mechanisms to ensure consistency across frames, in case segmented objects temporarily fail to detect (Fig. 12).

First, we have a persistence mechanism that keeps track of the last frame index where each instance was seen. We keep the original mask of each frame for retrieval. We define a persistence parameter per, meaning an instance can remain "remembered" for per frames before being forgotten. If an instance is not found in the current frame and was seen in the past per frames, we can reintroduce it from the last known position.

We have also introduced weighted average of past avg frames to produce extra temporal consistency. First, we compute exponential weights:

$$w_i = e^{-\lambda i}, \quad i = avg - 1, avg - 2, \ldots, 0$$

where:w_i represents the weight assigned to the i-th frame. λ is the decay rate, determining how quickly older frames lose importance. i decreases from $avg - 1$ (current frame) to 0 (oldest frame). avg is the total number of frames in the averaging window. If we have observed less than avg frames so far, we slice weights we don't need. We now apply normalization and a thresholding step—pixels below $thresh$ are set to 0.

Through this approach, we can emphasize consistent instances across frames, prevent flickering. Consistent instances will receive higher weight during the averaging process. We can also reduce the impact of short-lived segmentation artifacts, because they will have low weight and will be

discarded by the thresholding step. We can also maintain an instance's presence when it temporarily disappears. In our implementation, for objects, $thresh = 0.1$, $avg = 5$, and $\lambda = 1.2$; for arms, $thresh = 0.3$, $avg = 10$, and $\lambda = 0.8$; for scenes, $thresh = 0.3$, $avg = 10$, and $\lambda = 1.0$. We determined these values through trial and error on our evaluation videos.

C. Depth-based Improvements

We introduced temporal consistency in depth estimation using TCMonoDepth [19], a model that takes consecutive RGB video frames and applies Temporal Consistency Loss (TC Loss) along with Optical Flow Estimation to ensure smooth and stable depth predictions across frames. Considering that we focus on indoor scenes, clipping the farthest depth values is unnecessary—unlike outdoor environments, where distant objects may introduce noise or be irrelevant, indoor scenes typically have limited depth, thus all objects within the scene are contextually important. The Fig. 13 below demonstrates that removing depth clipping reveals previously discarded parts of important objects, ensuring a more complete and accurate depth representation.

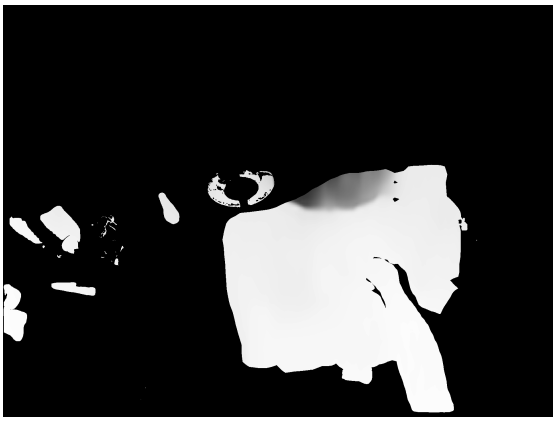

Fig. 13. Combination: 50% saliency no clipped quadratic (frame 11)

Hence, no clipping is used in our approach. TCMonoDepth in Figure 14 yields a noticeable improvement over Monodepth2 in Figure 5.

Fig. 14. Depth map from TCMonoDepth (frame 11)

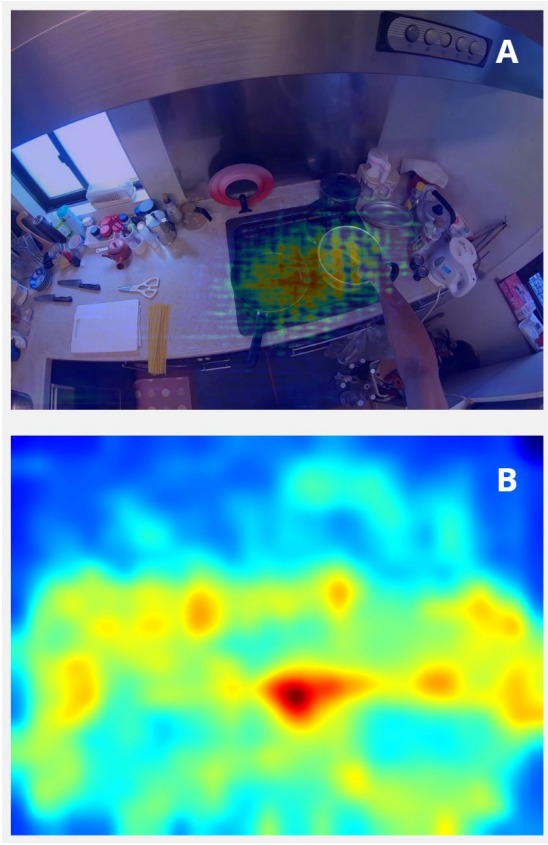

Fig. 15. Improved saliency map from DeepGaze III (frame 11). A) EgocentricGazePrediction predicted gaze heatmap. B) DeepGaze III saliency map using gaze from A and other previous frame gaze heatmaps.

D. Saliency-based Improvements

We introduce temporal consistency in saliency using DeepGaze III [16], which extends DeepGaze II by incorporating past fixations to modulate the next fixation probability heatmap. This is different from the static saliency by DeepGaze II, which detects visually striking areas based on low-mid level features per-frame. Here, DeepGaze III can produce dynamic saliency based on higher-level semantic patterns, predicting realistic human gaze behavior with smooth scan paths. Hence, it does not highlight important regions but models gaze dynamics, making it unsuitable to aid segmentation. And be used solely as scene simplification strategy.

To determine the fixation for each frame and pass this in to the fixation history for DeepGaze III, we use another model, Egocentric Gaze Prediction [13], that uses task-dependent attention transitions to determine the gaze heatmap. We compute the center of mass for this heatmap and use this location as the fixation in DeepGaze III as shown in Fig. 15.

E. Combination

We introduce four combinations based on our pipeline improvements:

1) Baseline scheme, temporal models: We implement the same relationship between saliency, segmentation, and depth as in the baseline, but with the introduced temporal models.

We use original segmentation output from DEVA, logical OR with DeepGaze III (top 5%) and weight with depth map from TCMonoDepth.

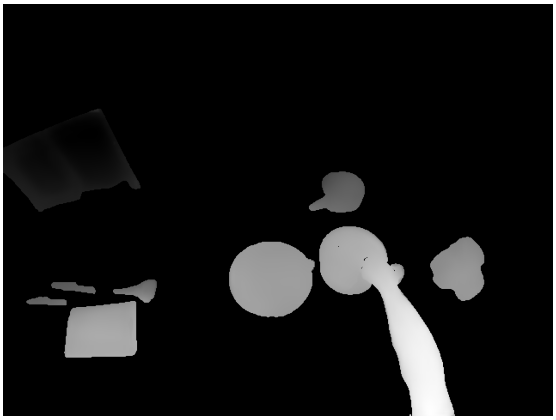

Fig. 16. Baseline scheme, temporal models (frame 11)

2) Depth-driven scheme, temporal models: Similar to the previous combination, we use original segmentation output from DEVA and weight with the depth map from TCMonoDepth. As we can see from the differences, the most salient region is gaze and it no longer reflects low-level interest regions, proving that it is not suitable to aid segmentation. We will talk about this more in results.

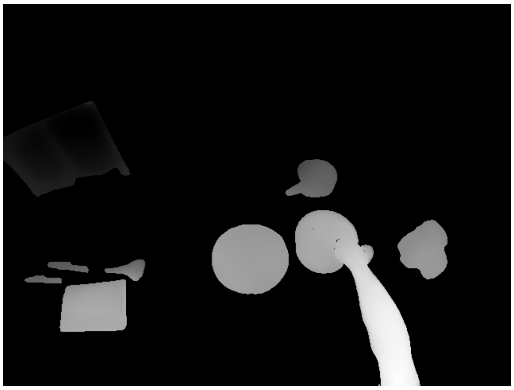

Fig. 17. Depth-driven scheme, baseline scheme without saliency (frame 11)

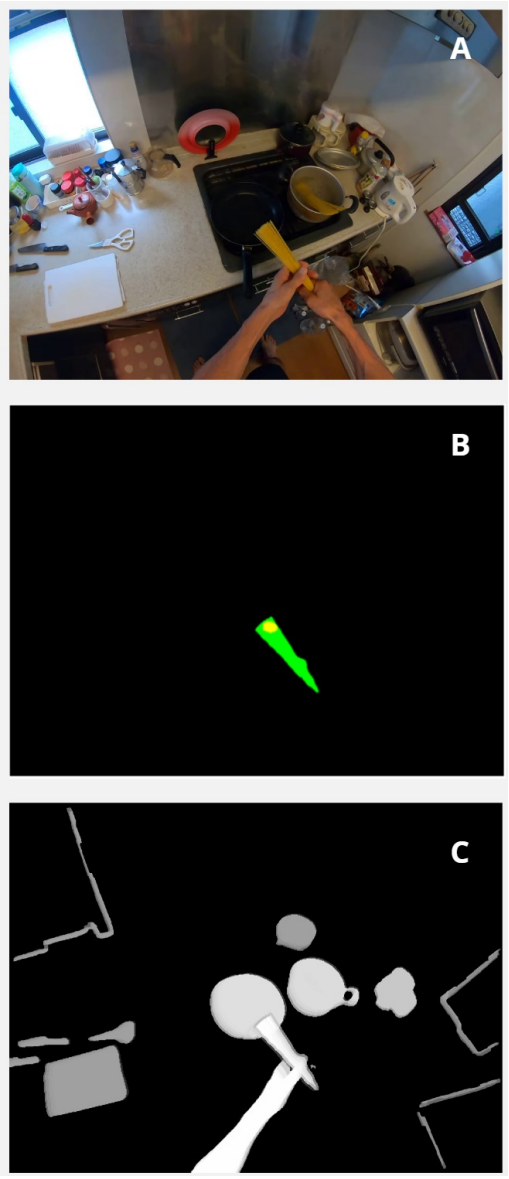

Fig. 18. Saliency-driven scheme, temporal models (frame 142). A) pure RGB frame. B) instance intersection with saliency. C) Full priority table implementation, with persistence and averaging.

3) Saliency-driven scheme, temporal models: Rather than using the saliency as segmentation aid. We complete the implementation of primary objects in the priority table, each object instance are "near gaze" if any of the pixels intersect the threshold salient region (in our implementation we retain only the 5% most salient pixels). The near gaze object will be assigned brightness b1, and its edges will be slightly reduced in brightness by a factor of 1.159. We maintain a finite history (past 10 frames) of the saliency and maintain the most recent one. This ensures that if the saliency is temporarily undetected due to model errors, the algorithm can refer to recent frames to infer continuity.

4) Baseline scheme, temporal models with single segmentation prompt: Here, we implement the same baseline relation-

ship between saliency, segmentation, and depth, and do not use the priority table improvements. Instead, the modification we make is to prompt DEVA once for all kitchen-related segments. In previous iterations, we would specify a scene prompt, object prompt, and arm prompt, which would be more selective in the objects it segments.

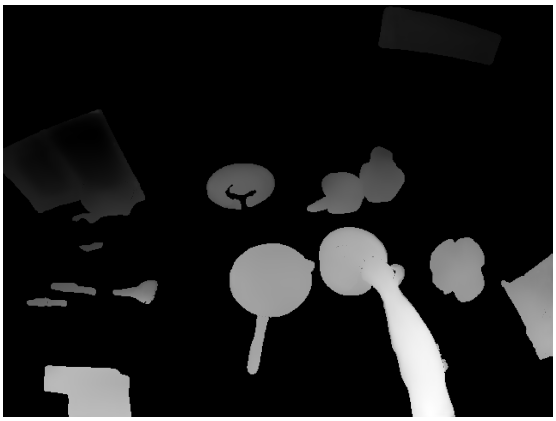

Fig. 19. Baseline scheme, temporal models with single segmentation prompt (frame 11)

IV. Evaluation

Due to compute and storage limitations, our evaluation of suggested improvements is composed of primarily qualitative analysis. First, when fine-tuning our improvements, we relied on kitchen-oriented tasks through the Ego4D dataset [10]. After producing the intended modifications to our pipeline, we transitioned to benchmark datasets that aligned more to our models' capabilities (depth, segmentation). We experimented on a few primary datasets:

- NYU Depth V2 [21] which contains 1.5K densely labeled RGB and depth pairs
- EgoDepthNormal [6] which contains >500K sychronized RGB-D frames, surface normal, and gravity vectors, specifically for egocentric video
- EPIC-KITCHENS VISOR [5] which contains >272K semantic masks of pixel-level object annotations, specifically for egocentric video

Moreover, we further qualitatively evaluate our temporal combinations by simulating prosthetic vision. We pass in this preprocessed video as an input stimulus to the pulse2percept open-source library [3] to provide a simulated representation of the physiological phenomena observed in the retina in [2], among others.

V. Results

Quantitatively, we can express some performance metrics from the static and temporal models. Note that we did not extract these ourselves—since we used off-the-shelf models, we simply display the information reported and cross-compare. First, in Table 2 we have depth estimation metrics between TCMonoDepth and variants of Monodepth2. TCMonoDepth is trained and tested on NYU Depth V2, which is composed of indoor video sequences and thus can be analyzed temporally. On the other hand, Monodepth2 is trained and tested on KITTI [8] which uses outdoor driving videos and collects data with laser scanning and GPS localization—this dataset is less relevant for our purposes. Regardless, considering limitations in compute and the fact that these two models were tested on different datasets, TCMonoDepth achieves higher scores than any variant of Monodepth2 on the listed metrics. For saliency and segmentation, we were unable to cross-evaluate any quantitative metrics.

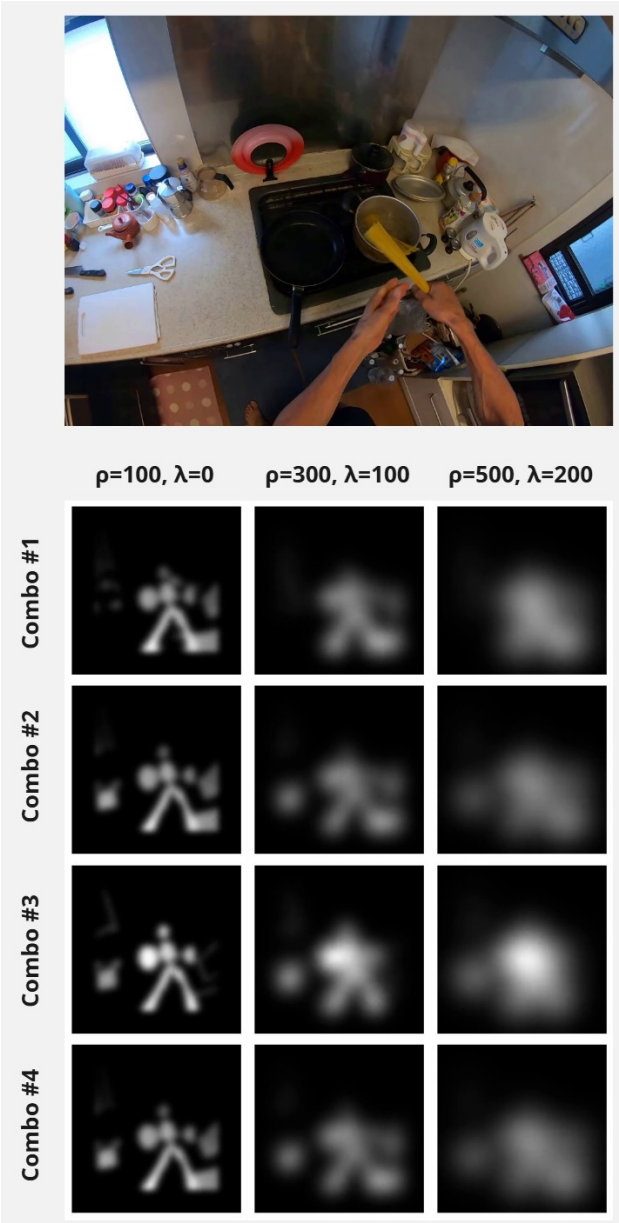

Fig. 20. Sample percepts varying retina-model parameters (frame 150)

Qualitatively, we evaluate our aforementioned temporal combinations in simulation. Following the baseline implementation, for each combination, we generated videos with retinal grid sizes $g \in \{8 \times 8, 16 \times 16, 32 \times 32\}$ and nine combina-

TABLE II
QUANTITATIVE METRICS FOR DEPTH ESTIMATION MODELS

Model	Resolution	Dataset	Abs Rel ↓	Sq Rel ↓	$\delta < 1.25 \uparrow$	$\delta < 1.25^2 \uparrow$	$\delta < 1.25^3 \uparrow$
Monodepth2 small	416×128	KITTI	0.118	0.935	0.852	0.949	0.976
Monodepth2 med (baseline)	640×192	KITTI	0.106	0.818	0.874	0.957	0.979
Monodepth2 large	1024×320	KITTI	0.106	0.806	0.876	0.958	0.980
TCMonoDepth (temporal)	448×320	NYU Depth V2	**0.1024**	**0.0556**	**0.9061**	**0.9808**	**0.9954**

tions of retinal-model parameters: $\rho \in \{100, 300, 500\}$ and $\lambda \in \{0, 100, 200\}$. Figure 20 shows sample frames from each combination.

This kind of analysis certainly points to a user study which we point to in future works. When varying the grid sizes, it became impossible to discern the task at hand from the 8×8—details became more visible at 16×16 and very comprehensible by 32×32 with low enough parameters. As we mentioned before, removing the saliency in combination 2 had very little effect on the baseline scheme, especially after the video had undergone substantial blurring. We believe the third combination to be the most effective out of these four for scene simpification—the edge markings we faint but present with low enough parameters, which helped make out object shapes. Moreover, combination 4 seemed to extract too many objects given its generic segmentation prompt, which as a result contributed to more blur and incoherence as the parameters increased. Overall, however, all of these combinations performed considerably better than the pure baseline, and reduced flickering by a substantial amount.

VI. DISCUSSION

The results of our study indicate notable advancements in scene simplification for bionic vision, specifically within egocentric indoor environments. Looking back, we suggested a significant number of improvements, and carefully choosing approaches (as well as hyperparameters) for saliency, segmentation, and depth allowed us to address key limitations of prior static models. Going forward, being more selective in our models and improvements will contribute to a more stable and informative visual representation, which is crucial for visual prostheses.

As we dealt mainly with qualitative analysis and evaluated our methods as we proceeded, a lot of our results were expected. Nevertheless, we did encounter a few exciting results for future work. For example, we noticed that segmentation models are pivoting toward automatic masking via natural language prompts. Both DEVA [4] and MESA (Matching Everything by Segmenting Anything) [25] are newer models that extract image areas with implicit semantic, and as a result are able to segment based on more semantic directives rather than hard-coded classes. Natural language prompt in scene simplification can allow user-specified preferences. This is an interesting direction to take visual neuroprostheses, as it could eliminate the need to determine which masks to obtain based on the user's context—instead, an audio queue ("I want to see ...") could be explored to guide the scene simplification approach. Integrating new modalities of input like this is also an exciting direction to support the pipeline. Moreover, not only segmentation, saliency can also be dynamically adjusted with extra modalities. For example, current state-of-the-art work in egocentric gaze anticipation takes audio signals along with the video input to predict future gaze, [18]. In this case gaze shift will be more temporally consistent and gaze predictions will be more context-aware and task-related. Moreover, innovations in 3D active objects tracking with egocentric camera [22] can enable maintaining awareness of object locations, even when they are not in the immediate field of view. With integration of prompts (e.g. query for object locations), it has great potential to further enhance adaptability and responsiveness of the system.

Moreover, expanding this pipeline to work in real-time is another logical next step—configuring these models to work in parallel, together with handling multiple stages of denoising, extracting edges, and dilating is a challenging endeavor. Lastly, a user study seems highly necessary given our limited amount of quantitative feedback—testing this modified pipeline on virtual patients and focusing on the percept aspect is another worthy next step.

Short project duration aside, we encountered a few limitations. For one, we were hindered by compute and storage—any video over 5 minutes with a high enough fps to conduct temporal analysis is at least 5GB, and testing any of the aforementioned models on a large depth or segmentation training set is out of the question. Additionally, we were blocked with the evaluation phase, since we tried to extract metrics for our baseline and temporal models on depth and segmentation benchmarks, but ran into issues with each benchmark we tried to implement (we could not extract complete depth data from EgoDepthNormal, depth metrics were suspiciously low across models for NYU Depth V2, EPIC KITCHENS VISOR data was too sparse and extracting dense annotations was unsuccessful). Regardless, we believe we have a substantial amount of content to show for our efforts.

VII. AUTHOR CONTRIBUTIONS

Yanxiu:

- **Research topic**: Scene simplification
- **Prior Research**: TCMonoDepth [19]; DEVA [4]; Egocentric Gaze Prediction[13]; MESA [25] ; egocentric gaze anticipation [18]; 3D active objects tracking [22]
- **METHODS**:
  - Baseline
  - Segmentation-based Improvements:

    - ∗ Baseline Improvements
    - ∗ Assigning Brightness on Priority
    - ∗ *(Extra) Temporal Segmentation Model*: Reviewed literature and identified deployable models, proposed usage of DEVA
    - ∗ *(Extra) Persistence and Averaging*: Integrate into our pipeline
  - Depth-based Improvements
  - *(Extra) Saliency-based Improvements*: Proposed usage of Egocentric Gaze Prediction as fixation
  - Combination
- **Experiment**: Experiment with Ego4D dataset and find optimal values for all parameters mentioned in METHOD. Produced 14 videos corresponding to different figures mentioned in the paper for further evaluations.
- **Paper writing**:
  - Methods
  - *(Extra) Discussions*: Future work in egocentric gaze anticipation and 3D active objects tracking.

Ron:

- **Research topic**: Temporal consistency; indoor scenes; Ego4D Dataset; Motivations
- **Prior Research and trials**: Literature Review; Deepgaze III [16]; tried Temporally Consistent Referring Video Object Segmentation with Hybrid Memory [20] and Temporally Consistent Depth Estimation Model[14]
- **METHODS**:
  - Segmentation-based Improvements:
    - ∗ Temporal Segmentation Model
    - ∗ Persistence and Averaging Across Frames
  - Saliency-based Improvements
  - *(Extra) Combination: Baseline scheme, temporal models with single segmentation prompt*: Provided prompted mask.
- **Experiment**: Experiment with NYU Depth V2, EgoDepthNormal and EPIC-KITCHENS VISOR datasets. Use p2p to simulate prosthetic vision with our temporal combinations (videos).
- **Evaluation**: Quantitatively evaluate performance on different datasets; Qualitatively evaluate our results on p2p.
- **Paper writing**:
  - Introduction
  - Related Work
  - *(Extra) Methods*: Provided Fig 8 and 15; Reformat passage structure and images; checked and refined contents; provide references.
  - Evaluation
  - Results
  - Discussions

REFERENCES

[1] A. K. Ahuja et al. "Factors Affecting Perceptual Threshold in Argus II Retinal Prosthesis Subjects". In: *Translational Vision Science & Technology* 2.4 (2013), p. 1. DOI: 10.1167/tvst.2.4.1. URL: https://tvst.arvojournals.org/article.aspx?articleid=2120952.

[2] Michael Beyeler et al. "A model of ganglion axon pathways accounts for percepts elicited by retinal implants". In: *Scientific Reports* 9.1 (2019), p. 9199. DOI: 10.1038/s41598-019-45416-4. URL: https://www.nature.com/articles/s41598-019-45416-4.

[3] Michael Beyeler et al. "pulse2percept: A Python-based simulation framework for bionic vision". In: *Proceedings of the 16th Python in Science Conference (SciPy)*. 2017, pp. 81–88. DOI: 10.25080/shinma-7f4c6e7-00c. URL: https://bionicvisionlab.org/publications/2017-pulse2percept/.

[4] Ho Kei Cheng et al. "Tracking Anything with Decoupled Video Segmentation". In: *ICCV*. 2023.

[5] Ahmad Darkhalil et al. *EPIC-KITCHENS VISOR Benchmark: VIdeo Segmentations and Object Relations*. 2022. arXiv: 2209.13064 [cs.CV]. URL: https://arxiv.org/abs/2209.13064.

[6] Tien Do, Khiem Vuong, and Hyun Soo Park. "Egocentric Scene Understanding via Multimodal Spatial Rectifier". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022.

[7] Monika Fleckenstein, Steffen Schmitz-Valckenberg, and Usha Chakravarthy. "Age-Related Macular Degeneration: A Review". In: *JAMA* 331.2 (2024), pp. 147–157. DOI: 10.1001/jama.2023.26074. URL: https://pubmed.ncbi.nlm.nih.gov/38193957/.

[8] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? The KITTI vision benchmark suite". In: *2012 IEEE Conference on Computer Vision and Pattern Recognition*. 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[9] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. "Digging Into Self-Supervised Monocular Depth Estimation". In: *CoRR* abs/1806.01260 (2018). arXiv: 1806.01260. URL: http://arxiv.org/abs/1806.01260.

[10] Kristen Grauman et al. "Ego4D: Around the World in 3, 000 Hours of Egocentric Video". In: *CoRR* abs/2110.07058 (2021). arXiv: 2110.07058. URL: https://arxiv.org/abs/2110.07058.

[11] Nicole Han et al. *Deep Learning–Based Scene Simplification for Bionic Vision*. 2021. arXiv: 2102.00297 [cs.CV]. URL: https://arxiv.org/abs/2102.00297.

[12] Stephen L. Hicks et al. "A Depth-Based Head-Mounted Visual Display to Aid Navigation in Partially Sighted Individuals". In: *PLOS ONE* 8 (July 2013), pp. 1–8. DOI: 10.1371/journal.pone.0067695. URL: https://doi.org/10.1371/journal.pone.0067695.

[13] Yifei Huang et al. "Predicting gaze in egocentric video by learning task-dependent attention transition". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 754–769.

[14] Numair Khan et al. *Temporally Consistent Online Depth Estimation Using Point-Based Fusion*. 2023. arXiv: 2304.07435 [cs.CV]. URL: https://arxiv.org/abs/2304.07435.

[15] Alexander Kirillov et al. "Segment Anything". In: *arXiv:2304.02643* (2023).

[16] Matthias Kümmerer, Matthias Bethge, and Thomas S. A. Wallis. "DeepGaze III: Modeling free-viewing human scanpaths with deep learning". In: *Journal of Vision* 22.5 (2022), p. 7. DOI: 10.1167/jov.22.5.7. URL: https://doi.org/10.1167/jov.22.5.7.

[17] Matthias Kümmerer, Thomas S. A. Wallis, and Matthias Bethge. "DeepGaze II: Reading fixations from deep features trained on object recognition". In: *CoRR* abs/1610.01563 (2016). arXiv: 1610.01563. URL: http://arxiv.org/abs/1610.01563.

[18] Bolin Lai et al. *Listen to Look into the Future: Audio-Visual Egocentric Gaze Anticipation*. 2024. arXiv: 2305.03907 [cs.CV]. URL: https://arxiv.org/abs/2305.03907.

[19] Siyuan Li et al. "Enforcing Temporal Consistency in Video Depth Estimation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 2021.

[20] Bo Miao et al. *Temporally Consistent Referring Video Object Segmentation with Hybrid Memory*. 2024. arXiv: 2403.19407 [cs.CV]. URL: https://arxiv.org/abs/2403.19407.

[21] Pushmeet Kohli Nathan Silberman Derek Hoiem and Rob Fergus. "Indoor Segmentation and Support Inference from RGBD Images". In: *ECCV*. 2012.

[22] Chiara Plizzari et al. *Spatial Cognition from Egocentric Video: Out of Sight, Not Out of Mind*. 2025. arXiv: 2404.05072 [cs.CV]. URL: https://arxiv.org/abs/2404.05072.

[23] Alex Rasla and Michael Beyeler. *The Relative Importance of Depth Cues and Semantic Edges for Indoor Mobility Using Simulated Prosthetic Vision in Immersive Virtual Reality*. 2022. arXiv: 2208.05066 [cs.HC]. URL: https://arxiv.org/abs/2208.05066.

[24] Yuxin Wu et al. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[25] Yesheng Zhang and Xu Zhao. *MESA: Matching Everything by Segmenting Anything*. 2024. arXiv: 2401.16741 [cs.CV]. URL: https://arxiv.org/abs/2401.16741.

[26] Bolei Zhou et al. "Semantic Understanding of Scenes through the ADE20K Dataset". In: *CoRR* abs/1608.05442 (2016). arXiv: 1608.05442. URL: http://arxiv.org/abs/1608.05442.