Scene Specific Navigation using CLIPSeg

Group Members: Aditya Potnis (apotnis2), Aidan Wefel(awefel2), Sai Aitha (sa60)

Table of contents

Motivation

Recent advances in open-vocabulary scene understanding and segmentation have significantly improved the ability of vision models to interpret complex, unstructured environments. Self-supervised learning models like DINOv2 [Oquab et al.] , contrastive models like CLIP [Radford et al., 2021] and open vocabulary segmentation models like CLIPSeg [Lüddecke et al., 2022] have demonstrated strong performance in open-ended object recognition and segmentation without requiring extensive task-specific labeling.

Motivated by these developments, we aim to design an integrated system that leverages these state-of-the-art techniques – scene understanding, open-vocabulary segmentation, and monocular depth estimation – for goal-driven navigation in unstructured environments. Our system has potential applications in robotic navigation in diverse and dynamic environments, as well as assistive technologies for the visually impaired to navigate unfamiliar spaces using semantic and spatial cues.

Current monocular navigation models like the “Obstacle Avoidance Strategy for Mobile Robot Based on Monocular Camera [T. Dang et al. (2023, *Electronics*)]” uses a single Fully Convolutional Network with a VGGNet backbone for the depth estimation network across different spaces, leading to metric scale drift, while other models do not recognize and assign risk to objects based on semantics. Our approach also allows for better scene detection and segmentation since it uses more fine tuned models which were not previously used. The design of “Monocular Based Navigation System for Autonomous Ground Robots Using Multiple Deep Learning Models [Z.. Machkour Et. Al.]” uses a Mask R-CNN for segmentation with which can perform segmentation with a one-hot vector, which makes it difficult to segment objects which may have more than one name or have vague identifiers such as “boot” and “footwear” which CLIP is able to identify much better. Approaches like CoNVOI (Context-aware Navigation using Vision Language Models in Outdoor and Indoor Environments) [A. J. Sathyamoorthy *et al.*] use a VLM for social specific and CLIP for scene based prompt selection, but utilize a high cost LIDAR for depth estimation for the map generation, which leads to high overall cost of the system.

We propose a modular architecture combining (i) CLIP for high-level scene understanding and selection of an appropriate monocular depth estimation model for metric distances (ii) Depth Anything V2 for scene specific metric depth estimation (iii) CLIPSeg for open-vocabulary semantic segmentation to identify goal objects and risky obstacles. (iv) A point cloud to grid map conversion with noise filtering and rasterization (v) A*-based path planning algorithm to compute a navigable path to the goal object.

We evaluate the robustness of each subcomponent individually with a variety of datasets and demonstrate the effectiveness of the integrated system through a variety of test cases.

Background

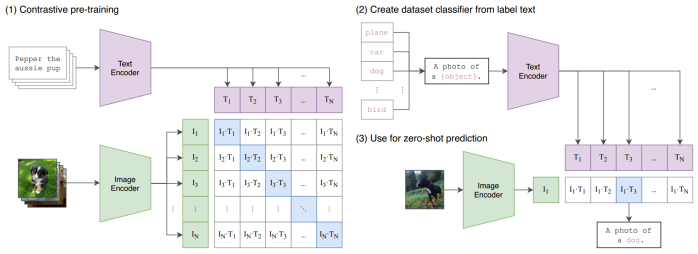
Modern visual-language navigation requires the following components for navigating spaces. These components can be a combination of the listed subcomponents, but generally are found in this form in some way. (1) domain-conditioned segmentation, (2) depth estimation

through active or passive sensing, (3) occupancy grid projection, and (4) heuristic or probabilistic graph search for path planning. Below we outline the ingredients in our system and the background concepts our system relies on.

1) Contrastive vision–language pre-training (CLIP)

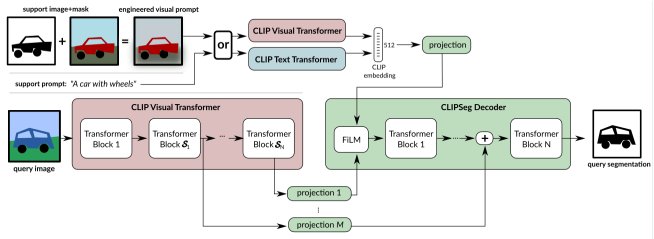
Radford et al. show that pairing 400 M (image, text) samples and training a model to match captions to images yields embeddings that let natural language become a task prompt. Their CLIP model transfers to 30+ vision datasets with no task-specific labels, establishing the idea of zero-shot scene classification we use to pick the right depth model checkpoint.

Contribution to our system: CLIP's cosine similarity scores between the frame and the text prompts "indoor" vs. "outdoor" drive our depth-model selector. [18]



2) Open-vocabulary segmentation (CLIPSeg)

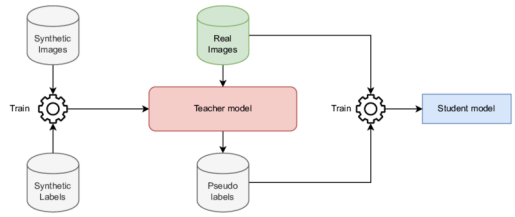
Lüddecke & Ecker plug a lightweight transformer decoder onto CLIP's frozen encoder so that a single text prompt (e.g., "fire extinguisher") produces pixel-wise masks at up to 15 fps. [19] This lets us tag any goal or obstacle without re-training.



3) Metric monocular depth (Depth Anything V2)

Presents a three-stage recipe—synthetic pre-training, large-teacher distillation, and metric fine-tuning—that yields fast, accurate monocular depth at multiple model sizes. The depth quality scales with synthetic teacher size and pseudo-label breadth. This achieves best in class performance for depth estimation.

Contribution to our system: The indoor/outdoor checkpoints give low-drift metric depth from a single RGB frame, enabling LiDAR-free grid-maps.This supplies the metric depth estimates that replace LiDAR in our pipeline; its indoor/outdoor checkpoints minimise scale drift when switched via CLIP scores. [3]



4) Point-cloud , and grid map rasterization (Open3D)

We back-project the depth map with the Open3D library, to fuse, filter, and down-sample point clouds in real time [11]. The cloud is voxelized into a 2-D cost grid for planning. This allows to project the environment information into a denser grid map which is sufficient for terrestrial locomotion as overhead objects do not pose a threat for the viewer / robot.

$$s\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

2D Image Coordinates | Intrinsic properties (Optical Centre, scaling) | Extrinsic properties (Camera Rotation and translation) | 3D World Coordinates

5) Heuristic path search (A*)

Hart et al.'s A* algorithm expands nodes in cost + heuristic order, guaranteeing the shortest path when the heuristic is admissible. We modify the termination rule to stop on any pixel in the goal mask.

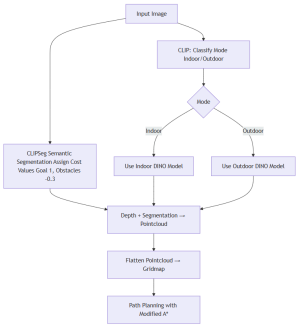
Key technological innovation

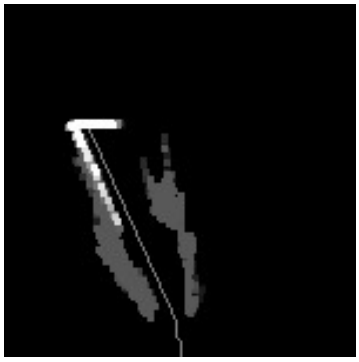
We introduce a novel system design for scene detection with a single RGB-frame navigation stack. The stack chooses the best depth estimator for a scene using CLIP, segments risk and goal regions with CLIPSeg, calculates metric depth using monocular depth estimation without need of expensive hardware. It then converts depth to a sparse point cloud, rasterizes it into a local 2-D grid-map, and runs a modified A* planner that knows which high-value pixels (goal mask) it may terminate on, even if the direct centroid is occluded.

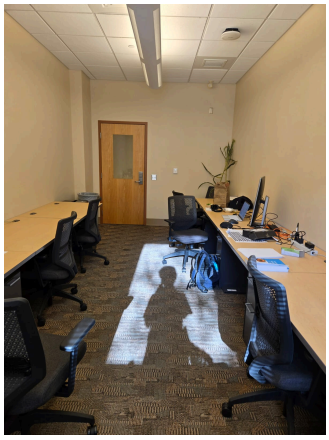
To our knowledge, the combination of adaptive depth-model switching plus mask-aware navigation has not appeared in previous systems. Most approaches either fix one monocular depth network (BehAV[K. Weerakoon *et al.*]) or rely on expensive LiDAR (CoNVOI[A. J. Sathyamoorthy *et al.*]) and multi-frame mapping (VLMaps[S.Taguchi Et. al]).

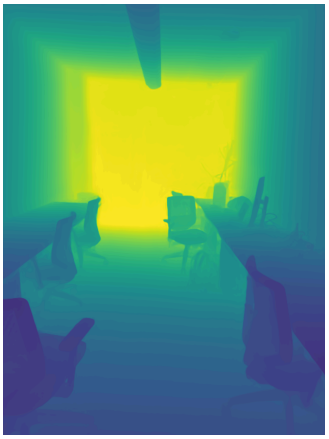
Unlike prior implementation in "Obstacle Avoidance Strategy for Mobile Robot Based on Monocular Camera"[Xun Yuan Et. Al (IROS2020)] that hard-code a single FCN + VGGNet depth network or closed-set segmentation, ours swaps a better fine tuned ViT monocular depth model Depth-Anything V2 indoor or outdoor model based on CLIP's "indoor/outdoor" classification, giving robust performance across lighting and context.

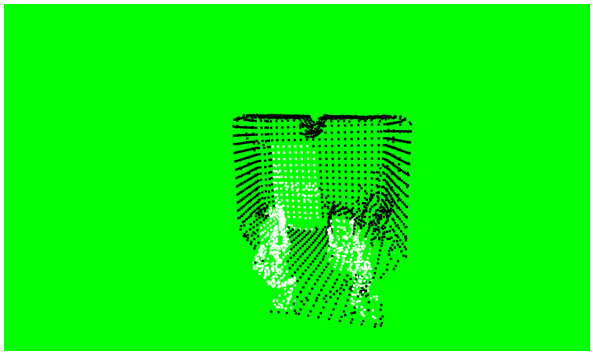
We were inspired by the CoNVOI approach, which uses CLIP for scene detection for different LLM prompts to navigate semi structured spaces. While this approach is robust and reliable at indoor/outdoor navigation, the approach requires a LIDAR (Light Detection and Ranging). This leads to the system becoming expensive. Our approach allows us to keep the cost low by using monocular estimation to keep system cost low. We also do not require to generalize the model for different spaces, this helps to make the model training specific to a specific domain since Depth-Anything V2 provides metric scale indoors and outdoors after a single fine-tune, unlike single-domain depth nets used in prior work, which can help make fine tuning costs for the model much lower.











Datasets

We use premade pix3d to validate the segmentation capability of clipseg. The depth data for indoor use was analyzed using the Matterport 3D and NYU Depth v2 datasets. For outdoor performance analysis, we used the KITTI Raw dataset for performance analysis of monocular depth estimation using depth anything v2.

We also created a set of indoor and outdoor images from our apartments and siebel with ground truth measurement using a measuring tape to the goals objects at Siebel School of Computer Science.

Dataset	Size / Modality	Labels & Metadata	Licence	Test subset for grid map analysis
NYU Depth V2	795 train / 654 test RGB–D frames (640 × 480)	Metric depth, per-pixel semantic classes	MIT license	30 images
Pix3D	10 k images + CAD alignments	Bounding boxes & instance masks for 9 furniture classes	CC-BY-NC-SA	30 images
Matterport 3D	134 k RGB-D panoramas, 61 buildings	Global poses,rgb images, semantic meshes	Matterport Terms of Use	30 images
KITTI Raw	57 k outdoor stereo frames	Poses, LiDAR depth, odometry	CC BY-NC-SA	30 images
Our In-house photo set	20 indoor and 30 outdoor photos in and around the UIUC campus. 6 videos	Images, Tape-measured depth to chosen goal point	Internal use	20 images and 6 videos
Nvidia r2b_2023 dataset	8 robotics (ROSBAG) dataset converted to mp4 image and depth video in FPV	Poses, LIDAR, Depth, rgb image	CC BY 4.0	9 rosbags

Evaluation criteria

Our primary metric is **Path-Success Rate (PSR)**

$$\mathrm{PSR} = \frac{N_{\mathrm{succ}}}{N_{\mathrm{tot}}}$$

where $Nsucc$ is the number of test scenes in which the planner finds a collision-free path that reaches any pixel whose greyscale value exceeds 250 (goal mask), and $Ntot$ is the total number of scenes.

We used different metrics to measure similarity scores, the variables x, y are the embedding vectors of input and reference values respectively in R^n .

1) L2 / Euclidean distance:

$$d_{L2}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

2) Cosine similarity:

$$s_{\cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|_2 \, \|\mathbf{y}\|_2} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \, \sqrt{\sum_{i=1}^{n} y_i^2}}$$

3) Manhattan distance:

$$d_{L1}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$$

4) Dot-product similarity:

$$s_{\mathrm{dot}}(\mathbf{x}, \mathbf{y}) = \mathbf{x} \cdot \mathbf{y} = \sum_{i=1}^{n} x_i y_i$$

Clip Scene detection accuracy was measured by

$$\hat{y} \;=\; \arg\max\Big\{\, \mathrm{score}_{\mathrm{indoor}},\; \mathrm{score}_{\mathrm{outdoor}} \,\Big\}$$

where $\mathrm{score}_{\mathrm{indoor}}$ and $\mathrm{score}_{\mathrm{outdoor}}$ are cosine similarity scores in range of 0 to 1.
We reviewed the clip score for the percentage of correct classifications. For indoor and outdoor scenes.

$$\mathrm{Accuracy} \;=\; \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\big[\hat{y}^{(i)} = y^{(i)}\big]$$

$$\text{Indoor accuracy} = \frac{1}{N_{\text{in}}} \sum_{i:\, y^{(i)} = \text{indoor}} \mathbf{1}\big[\hat{y}^{(i)} = \text{indoor}\big]$$

$$\text{Outdoor accuracy} = \frac{1}{N_{\text{out}}} \sum_{i:\, y^{(i)} = \text{outdoor}} \mathbf{1}\big[\hat{y}^{(i)} = \text{outdoor}\big]$$

We used the pix3d dataset to evaluate CLIPSEG's performance, comparing the Mean squared error (MSE) and Similarity (1 - (sum of differences)/(pixels * 255)) of the generated masks with the ground truth masks

For the depth module we report **root-mean-square error (RMSE)** and accuracy under the conventional delta thresholds; RMSE is

$$\mathrm{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(d_i - \hat{d}_i)^2},$$

Where d_i and $\hat{d}_i$ are ground-truth and predicted depths at pixel i and M is the number of valid pixels.

Baselines and Evaluations results:

1) CLIP vs DINO comparison and Scene recognition analysis

CLIP v. DINOv2 embedding similarity and quality analysis

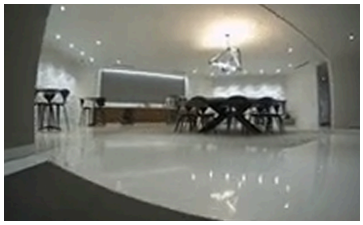
We initially compared different similarity score metrics across different image / videos from DINOv2 and CLIP to analyze quality of scene detection. We found in the comparison that the CLIP model had better scene understanding and was less affected by the spatial nature of the image embeddings.

We compared the L2 (euclidean), Manhattan distance, dot product and cosine similarity (normalized) scores across the 6 test out of distribution videos in indoor and outdoor environments. We used some videos recorded by us and some videos from the nvidia r2b_2023 robotics dataset. Similar to the research in literature, they found the confirmation of the cosine similarity being the most relevant metric for measuring similarity with embeddings.

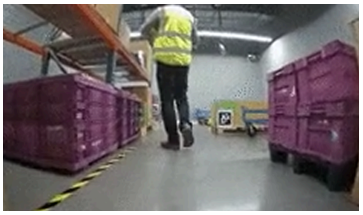
Compared to DINOv2 we found that CLIP had a smoother similarity distance profile with identical scenes having a small change in similarity score between the first frame and the consequent frames.

The table of images shows an example for the same. DINO had sudden changes when the scene changed slightly which is not ideal behaviour for the scene recognition test that we were conducting.

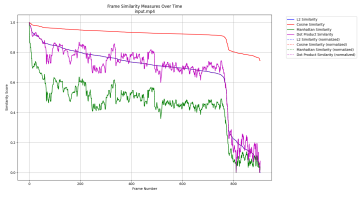
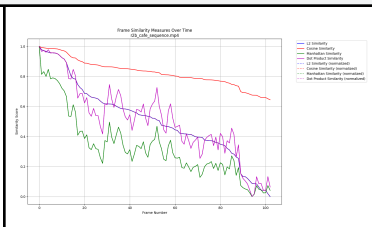
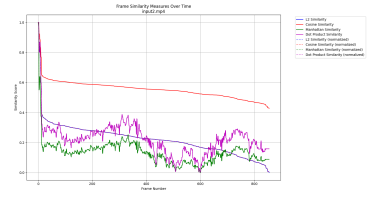
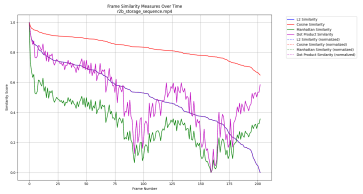
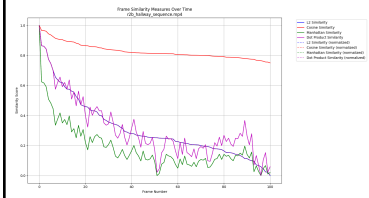
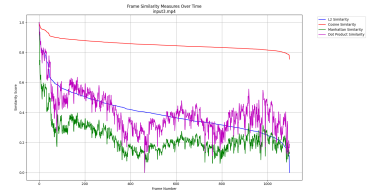
Hence we decided to use CLIP as the scene identifier model for the navigation module.

Sample test video frame grab, The test measures embedding feature space deviation using different metrics from the first frame image's feature embedding.		
		

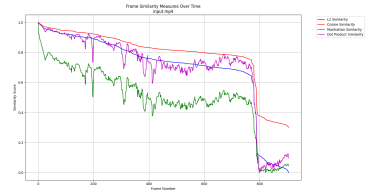
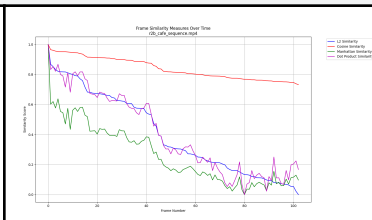
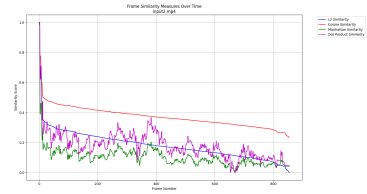
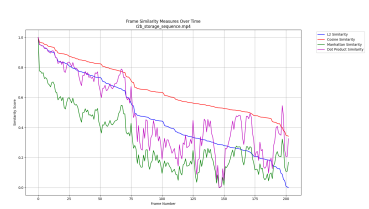
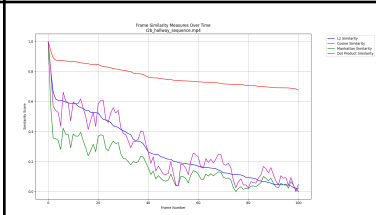
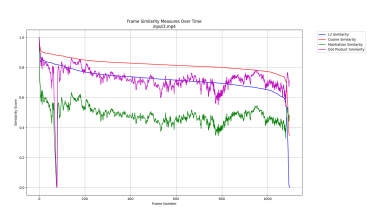
Sample test video frame grab, The test measures embedding feature space deviation using different metrics from the first frame image's feature embedding.



Framewise similarity score for CLIP in video with respect to first frame, cosine (red), L2 (blue), purple (dot product), green(manhattan distance)

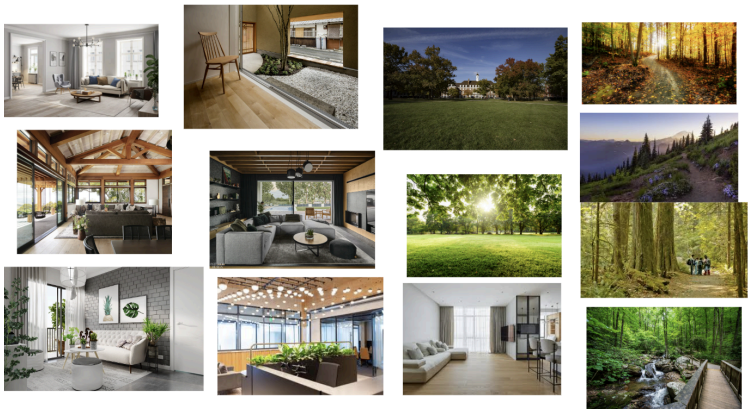







Framewise similarity score for DINOv2 in video with respect to first frame, cosine (red), L2 (blue), purple (dot product), green(manhattan distance)

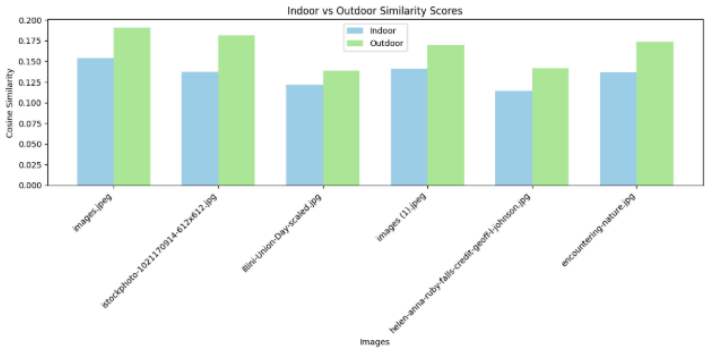
CLIP analysis for scene detection across indoor and outdoor images

We tested CLIP features from the text encoding for the prompts “indoor” and “outdoor” on a variety of image sets. These included data from the matterport 3d and some UIUC indoor outdoor image datasets. We also tested the argmax similarity score on a variety of images from our photo dataset to analyze performance on out of distribution data.



The sample dataset with indoor and outdoor photos from the pix3d and matterport datasets along with some images from the internet.

We found that on our datasets, the performance of the CLIP model was consistent at detecting scene cosine similarity. For indoor and outdoor spaces. On average the system was able to accurately identify scenes for 92.5% of our dataset. Most scores had a 30 to 40% difference between the “indoor” and “outdoor” prompt similarities. Some did have smaller differences in similarity score, but the top 1 similarity was correctly identified.



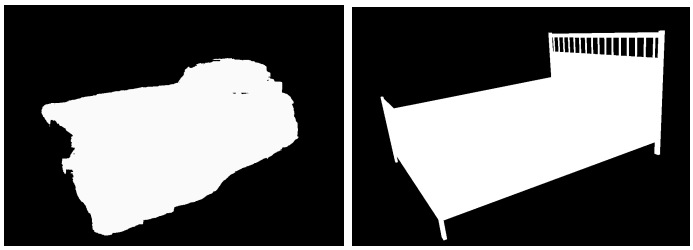
Outdoor images

We also noticed that certain objects in the scene led to the similarity score to drop such as indoor trees. One example is noted in the image below. We hypothesize that in the latent space, the embedding correlates “outdoor” keyword strongly with “trees”. We noticed a possible improvement this model can gain from by tuning the CLIP model better for the domain it is being used for. Overall we found CLIP to be robust at zero-shot out of distribution scene detection when lighting conditions are optimal, hence we decided to use it for our tests.

Dataset	Subset size for testing	Baseline accuracy	Results
Our test dataset	40 out of distribution images from various parts of UIUC campus	76.2% top-1 accuracy on ImageNet (zero-shot)	92.5% accuracy (37 scenes were correctly identified as indoor or outdoor)
Matterport 3D	100 image subset of randomly selected rgb indoor images		85% accuracy (85 scenes were correctly identified as indoor) Data classified as outdoor often had trees or other indoor plants in the image.

2) CLIPSeg segmentation analysis

We verify that CLIPSeg is working as intended in our system for common indoor objects. We use the Pix3d dataset with ~1000 images of indoor objects and masks for those objects. We generate masks using our system and compare them to the provided ground truth masks.



Mean Squared Error (MSE) is calculated by taking the difference between each pixel (0 - 255) and squaring it and taking the mean. Similarity is computed as 1 - (total difference) / (maximum possible difference)

Dataset	Mean Squared Error	Similarity
Pix3d - Overall	0.563	0.892

Interestingly, the tool dataset has the highest similarity. This could be because the tools take up less of the image, so there are less pixels to make mistakes on when making segmentation masks.

3) DepthAnythingV2

We evaluated Depth-Anything V2 on indoor scenes using the NYU Depth V2 dataset, which provides RGB–D image pairs for benchmarking monocular depth estimation. The model achieved an RMSE of 1.206 and an accuracy of 54% under the $\delta < 1.25$ threshold, improving to 95% under $\delta < 1.25^3$. While this shows reasonable performance, it falls short of state-of-the-art results on this dataset.

```
--- Final Evaluation Table ---
Metric       |  Value
----------------------
RMSE         | 1.206
MAE          | 0.947
AbsRel       | 0.407
δ < 1.25     | 0.539
δ < 1.25²    | 0.820
δ < 1.25³    | 0.949
```

The main reasons are that the model was not fine-tuned on NYU specifically, and our validation set likely lacked scene diversity. These results indicate that substantial fine-tuning is required for Depth-Anything V2 to match top performance on datasets like NYU Depth V2, especially for structured indoor environments.

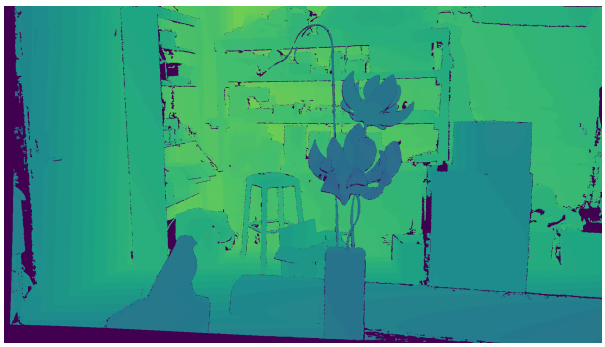
4) Analysis of Overall system performance after integration

We tested the model for different input images and checked the success rate at generating a path to an object set by a prompt in the scene. We found that most of the objects were identified correctly for our dataset. The Path Success Rate accuracy was 90%. The sample size of this test was 20 images. In the future we hope to study a larger sample size as it will give a better idea of the model's working. We also studied the accuracy of the path length generated by the model to the actual value to validate the system.

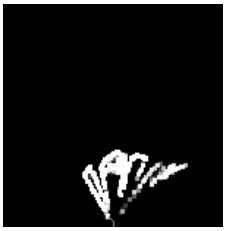
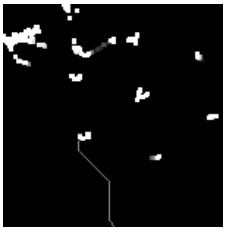
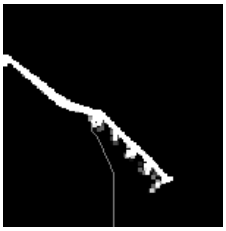
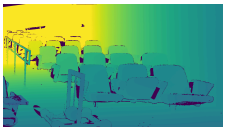
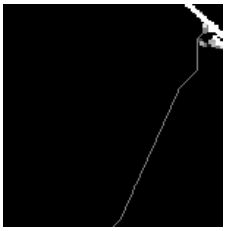
Dataset	Path Success rate	Path length average error w.r.t. actual length
Ours (UIUC photos)	90% (18 / 20)	9%

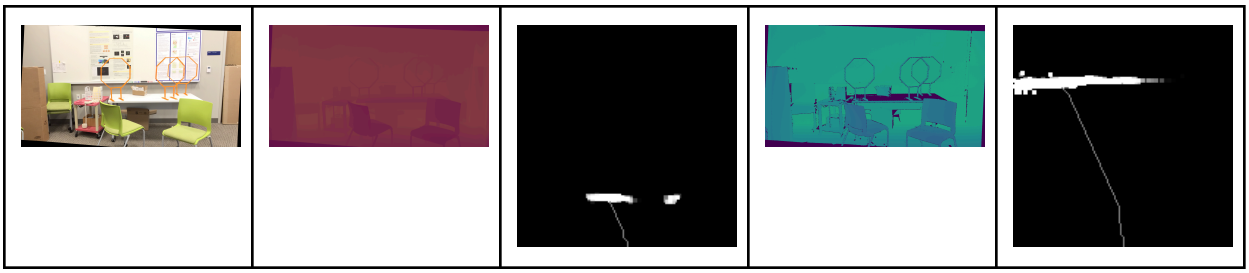
Ablation study

We evaluate the performance of using the DepthAnythingV2 model by using the Middlebury 2021 Stereo dataset[17]. We compare the output of our full pipeline with the output given by replacing the depth image created by DepthAnything with the provided depth information. In general, the depths provided with the dataset were more noisy than the generated depthmap. They also had a different scale, which led to different sized grid projections for the pathfinding.




We find that using the measured depth as provided by the Middlebury dataset does not significantly affect the quality of the output. Some examples are provided below. In order to always have a valid target for pathfinding to, we set the goal to “back wall”.

RGB image	DepthAnything	DepthAnything path	Measured	Measured path
				
				



Discussion

Our results show that it's possible to build a reliable navigation system using just a single RGB camera, without the need for expensive sensors like LiDAR. By combining CLIP for scene classification, DepthAnything for depth estimation, and CLIPSeg for segmentation, we were able to generate accurate point clouds and paths across both indoor and outdoor scenes.

The system reached its goal in 90% of test cases, and the predicted distances were within ±9% of the actual measured distances. However, we also found that lighting had a big effect on depth estimation and segmentation. In darker scenes or those with strange shadows, the system struggled more. Some small adjustments were also needed to match the estimated depths with real-world measurements, likely due to camera calibration issues.

Overall, our system proved to be strong at handling new scenes, even with some challenges. It supports the idea that combining scene understanding and depth estimation leads to better navigation, even without expensive hardware. For future work, we hope to add features that help the model identify safe areas to walk and possibly extend the planner to work in 3D instead of just 2D.

Looking ahead, there are several directions for improvement. We hope to make the system more robust to different lighting conditions by fine-tuning the models for specific environments. Adding the ability to identify safe or walkable areas in the scene would improve the quality of the paths generated, especially in cluttered environments. Extending the system to support full 3D path planning would allow it to handle elevation changes and obstacles more effectively. Lastly, incorporating temporal information across video frames could help in dynamic or moving environments, making the system even more adaptable.

Conclusion

Our single-RGB-frame navigation stack succeeds in combining open-vocabulary semantic segmentation and metric geometry into a light, low-cost pipeline that works across scene types. On a mixed indoor/outdoor photo set taken around UIUC, the integrated system reached its goal in 18 of 20 cases (90 % path-success rate) while keeping average path-length error within +/- 9% of hand-measured ground truth. Compared with a baseline that (i) fixes a single monocular depth network and (ii) stops only at the centroid of the goal object, our adaptive depth-switching and mask-aware A* together improved the PSR by 10% and reduced collision failures. Depth-Anything V2’s indoor/outdoor model checkpoints, chosen via CLIP

scene classification, limited metric-scale drift to ±9 % competitive with Stereo Vision and LiDAR systems yet achievable with low cost cameras.

The performance gains confirmed the hypothesis of combining tight coupling of semantic and depth based planning outperforms geometry-only or semantics-only approaches while avoiding the cost and power draw of active sensors. The stack was robust to modest occlusions, dynamic objects, and viewpoint shifts, achieving real-time inference on a laptop GPU.

The main hurdle we faced was due to lighting which caused degradation in both depth estimation and segmentation using clipseg. Future work may need fine tuning for various lighting conditions, extending the grid map to a 3d map for path planning in generalized space, using traversability cues to mark confidence of driving on surfaces.

Overall, we were able to verify that adaptive model selection and open vocabulary goal-mask assignment can be executed in a system to make an effective, scalable navigation model without expensive hardware for a variety of use cases ranging from robot navigation, image segmentation for vehicle repair to object identification for the visually impaired.

Statement of contributions

Aditya wrote first drafts for Motivation, Background, Key Technological Innovation, Datasets, Evaluation Criteria and conclusion. Aidan wrote first drafts for Baselines and Ablation study. Aditya wrote the code to evaluate the DINO vs CLIP models, and the integration code creating the pipeline of img -> CLIP -> Depth -> point cloud -> gridmap -> pathfinding. Aidan wrote the code used to evaluate the integration on various datasets. {mask comparison, indoor scenes, depth evaluation}. Sai wrote the code to integrate DepthAnything and also evaluated it with a broader dataset.

Bibliography

[1] K. Weerakoon et al., “BehAV: Behavioral Rule Guided Autonomy Using VLMs for Robot Navigation in Outdoor Scenes,” Oct. 02, 2024, arXiv:2409.16484. Accessed: Oct. 17, 2024. [Online]. Available: http://arxiv.org/abs/2409.16484

[2] A. J. Sathyamoorthy et al., “CoNVOI: Context-aware Navigation using Vision Language Models in Outdoor and Indoor Environments,” Mar. 22, 2024, arXiv: arXiv:2403.15637. Accessed: Nov. 04, 2024. [Online]. Available: http://arxiv.org/abs/2403.15637

[3] L. Yang et al., “Depth Anything V2,” Oct. 20, 2024, arXiv: arXiv:2406.09414. doi: 10.48550/arXiv.2406.09414.

[4] M. Oquab et al., “DINOv2: Learning Robust Visual Features without Supervision,” Feb. 02, 2024, arXiv: arXiv:2304.07193. doi: 10.48550/arXiv.2304.07193.

[5] T. Wolf et al., “HuggingFace’s Transformers: State-of-the-art Natural Language Processing,” Jul. 14, 2020, arXiv: arXiv:1910.03771. doi: 10.48550/arXiv.1910.03771.

[6] T. Lüddecke and A. S. Ecker, “Image Segmentation Using Text and Image Prompts,” Mar. 30, 2022, arXiv: arXiv:2112.10003. doi: 10.48550/arXiv.2112.10003.

[7] A. Radford et al., “Learning Transferable Visual Models From Natural Language Supervision,” Feb. 26, 2021, arXiv:2103.00020. doi: 10.48550/arXiv.2103.00020.
[8] M. Rey-Area, M. Yuan, and C. Richardt, “Matterport3D 360° RGBD Dataset.” University of Bath, p. README.md = 3kB, data_00.zip = 12GB, data_01.zip = 13GB, data_02.zip = 11GB, data_03.zip = 11GB, data_04.zip = 11GB, data_05.zip = 11GB, data_06.zip = 9GB, Mar. 25, 2022. doi: 10.15125/BATH-01126.
[9] Z. Machkour, D. Ortiz-Arroyo, and P. Durdevic, “Monocular Based Navigation System for Autonomous Ground Robots Using Multiple Deep Learning Models,” Int J Comput Intell Syst, vol. 16, no. 1, p. 79, May 2023, doi: 10.1007/s44196-023-00250-5.
[10] T.-V. Dang and N.-T. Bui, “Obstacle Avoidance Strategy for Mobile Robot Based on Monocular Camera,” Electronics, vol. 12, no. 8, p. 1932, Apr. 2023, doi: 10.3390/electronics12081932.
[11] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3D: A Modern Library for 3D Data Processing,” Jan. 30, 2018, arXiv: arXiv:1801.09847. doi: 10.48550/arXiv.1801.09847.
[12] X. Sun et al., “Pix3D: Dataset and Methods for Single-Image 3D Shape Modeling,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 2974–2983. doi: 10.1109/CVPR.2018.00314.
[13] C. Huang, O. Mees, A. Zeng, and W. Burgard, “Visual Language Maps for Robot Navigation,” in 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom: IEEE, May 2023, pp. 10608–10615. doi: 10.1109/ICRA48891.2023.10160969.
[14] NYU Depth V2: https://cs.nyu.edu/~fergus/datasets/nyu_depth_v2.html
[15] KITTI Raw: https://www.cvlibs.net/datasets/kitti/raw_data.php
[16] Nvidia r2b_2023 dataset: https://catalog.ngc.nvidia.com/orgs/nvidia/teams/isaac/resources/r2bdataset2023
[17] Middlebury 2021 Mobile stereo datasets https://vision.middlebury.edu/stereo/data/scenes2021/
[18] OpenAI CLIP https://openai.com/index/clip/
[19] CLIPSeg Mar. 30, 2022, arXiv: arXiv:2112.10003. doi: 10.48550/arXiv.2112.10003