2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

FreDSNet: Joint Monocular Depth and Semantic Segmentation with Fast Fourier Convolutions from Single Panoramas

Bruno Berenguel-Baeta and Jesus Bermudez-Cameo and Jose J. Guerrero

***Abstract*— In this work we present FreDSNet, a deep learning solution which obtains semantic 3D understanding of indoor environments from single panoramas. Omnidirectional images reveal task-specific advantages when addressing scene understanding problems due to the 360-degree contextual information about the entire environment they provide. However, the inherent characteristics of the omnidirectional images add additional problems to obtain an accurate detection and segmentation of objects or a good depth estimation. To overcome these problems, we exploit convolutions in the frequential domain obtaining a wider receptive field in each convolutional layer. These convolutions allow to leverage the whole context information from omnidirectional images. FreDSNet is the first network that jointly provides monocular depth estimation and semantic segmentation from a single panoramic image exploiting fast Fourier convolutions. Our experiments show that FreDSNet has slight better performance than the sole state-of-the-art method that obtains both semantic segmentation and depth estimation from panoramas. FreDSNet code is publicly available in `https://github.com/Sbrunoberenguel/FreDSNet`**

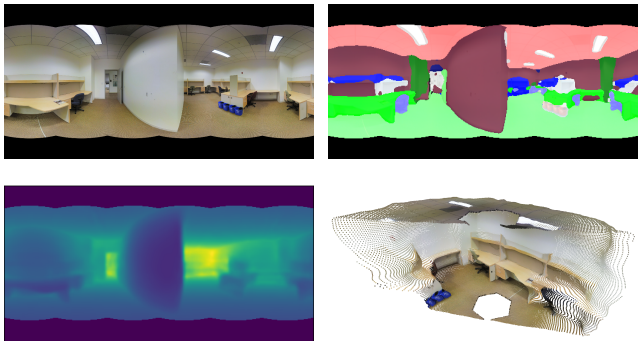

Fig. 1: Overview of our proposal. From a single RGB panorama (up left), we make a semantic segmentation (up right) and estimate a depth map (down left) of an indoor environment. With this information we are able to reconstruct in 3D the whole environment (down right).

I. Introduction

Understanding 3D indoor environments is a hot topic in computer vision and robotics research [1][2]. The scene understanding field has different branches which focus on different key aspects of the environment. The layout recovery problem has been in the spotlight for many years, obtaining great results with the use of standard and omnidirectional cameras [3][4][5][6]. This layout information is useful for guiding autonomous robots [7][8] or doing virtual and augmented reality systems. Another line of research focuses on detecting and identifying objects and their classes in the scene. There are many methods for conventional cameras [9][10][11], which provide great results, however conventional cameras are limited by their narrow field of view. In recent years, works that use panoramas, usually in the equirectangular projection, are increasing [12][13], providing a better understanding of the whole environment. Besides, the combination of semantic and depth information helps to generate richer representations of indoor environments [14][15]. In this work, we focus on obtaining, from equirectangular panoramas, two of the main pillars of 3D scene understanding: semantic segmentation and monocular depth estimation.

Without the adequate sensor, navigating autonomous vehicles in unknown environments is an extremely challenging task. Nowadays there is a great variety of sensors that provide accurate and diverse information of the environment (LIDARs, cameras, microphones, etc.). Among these possibilities, we choose to explore omnidirectional cameras, which have become increasingly popular as main sensor for navigation and interactive applications. These cameras provide RGB information of all the surrounding and, with the use of computer vision or deep learning algorithms, provide rich and useful information of the environment.

In this paper, we introduce FreDSNet, a new deep neural network which jointly provides semantic segmentation and depth estimation from a single equirectangular panorama (see Fig. 1). We propose the use of the Fast Fourier convolution (FFC) [16] to leverage the wider receptive field of these convolutions and take advantage of the wide field of view of 360 panoramas. Besides, we use a joint training of semantic segmentation and depth, where each task can benefit from the other. Semantic segmentation provides information about the distribution of the objects as well as their boundaries, where usually are hard discontinuities in depth. On the other hand, the depth estimation provides the scene's scale and the location of the objects inside the environment. With this information, we provide accurate enough information for applications as navigation of autonomous vehicles, virtual and augmented reality and scene reconstruction.

The main contribution of this paper is that FreDSNet is the first solution which jointly obtain semantic segmentation and depth estimation from single panoramas. For that, we include and exploit the FCC in a new network architecture for visual scene understanding. These convolutions have higher

All authors are with Instituto de Investigacion en Ingenieria de Aragon, I3A, Universidad de Zaragoza, Spain
Corresponding author: `berenguel@unizar.es`

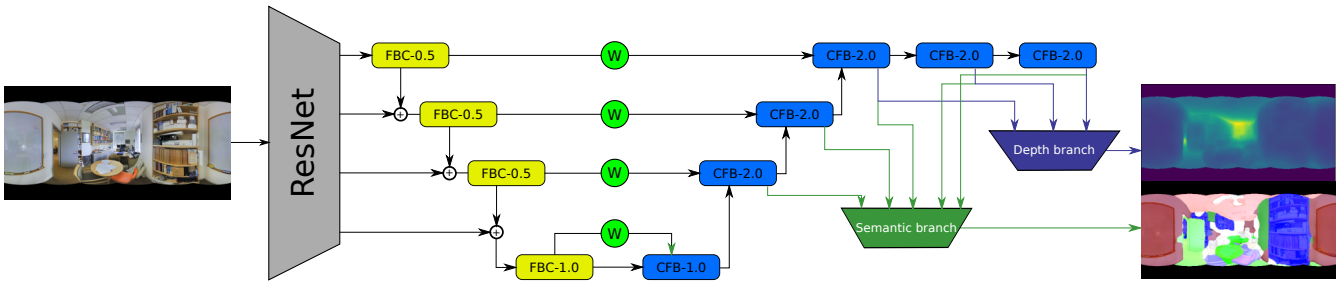

Fig. 2: Architecture of our **Fre**quential **D**epth estimation and **S**emantic segmentation **Net**work (FreDSNet). The encoder part is formed by a feature extractor (ResNet) and four encoder blocks. The decoder part is formed by six decoding blocks and two branches that predict depth and semantic segmentation. The skip connections from the encoder to the decoder use learned weights.

effective receptive field than standard convolutions, obtaining more context information in early layers. This is a key feature for our proposal since, being aware of the context improves the understanding of the scene and only with omnidirectional images we can obtain this information.

II. Related works

Semantic segmentation The semantic segmentation on perspective images is a well-studied field. We can find many works on object detection [11], semantic segmentation [10][17] or both tasks [9][18] from perspective cameras. However, omnidirectional images pose a harder problem which is more difficult to tackle. Then, only a few works are able to make semantic segmentation from omnidirectional images [12][19] and object detection [13]. Since omnidirectional images present heavy distortions (e.g. in spherical projections, like equirectangular images, this distortion is more accentuated in the mapping of the poles) these kinds of images are difficult to manually annotate. Nevertheless, due to the wide field of view of these images (e.g. in the spherical projection, we can see all the surroundings in a single image), the use of omnidirectional images in semantic segmentation is an active field of study since we can obtain a complete semantic understanding of the environment from a single image.

Depth estimation Monocular depth estimation is a research topic that has been on the spotlight in recent years. With the rise of deep learning methods, many works on depth estimation from conventional cameras have appeared with different approaches such as camera adapted convolutions [20]; convolutional networks with masking layers [21]; relative depth maps [22] or optical flow [23]. Almost at the same time, different works on depth estimation from panoramic images started to appear for indoor scene understanding purposes taking advantage of recurrent networks [24], attention mechanisms [19], twin networks [25] or convolutional networks [26]. Each work presents particular approaches for monocular depth estimation, being an open field of study with great interest and many applications.

Network architecture Many recent works on semantic segmentation or depth estimation rely on convolutional encoder-decoder architectures with some recurrent [24] or attention mechanism [19] as hidden representation of the environment. This kind of architectures aim to reduce the spatial resolution of the input image, increasing the number of feature maps in the encoder part, relating the general context of the environment in the hidden representation and up-sampling it in the decoder part to obtain the desired information. However, the traditional encoder-decoder architecture which relies on standard convolutions [26] or geometrical approximations [12] suffers from slow growth of the effective receptive field of the convolutions, losing the general context information that omnidirectional images provide.

In this work, we propose an encoder-decoder architecture for our network. However, we propose an adaptation of the fast Fourier convolution presented in [16], which we denominate Fourier Block since we modify the behavior of the block. These convolutions have proved that can 'see' the whole image at once, obtaining a higher effective receptive field from early layers which leads to a better understanding of omnidirectional images and the context information.

III. FreDSNet: monocular depth and semantic segmentation

Our network follows an encoder-decoder architecture, resembling U-net [27], with Resnet [28] as initial feature extractor and two separated branches for depth estimation and semantic segmentation. (see Fig.2). It is inspired by BlitzNet [9] and PanoBlitzNet [13], using multi-resolution encoding and decoding, in order to obtain a multi-scale representation of the scene, and the use of skip connections, which makes the training process more stable. Each branch takes intermediate feature maps from the decoder part to provide an output from the multi-scale decoded information. What differentiates our architecture is how the blocks of encoder and decoder are composed and how these parts are interconnected.

A. Architecture

The proposed encoder blocks (FBC-N) are formed by a Fourier Block (FB) followed by a down-scaling (N) and a set of standard convolutions (W-conv) as shown in Fig.3a. The Fourier Block has the same structure as the FFC

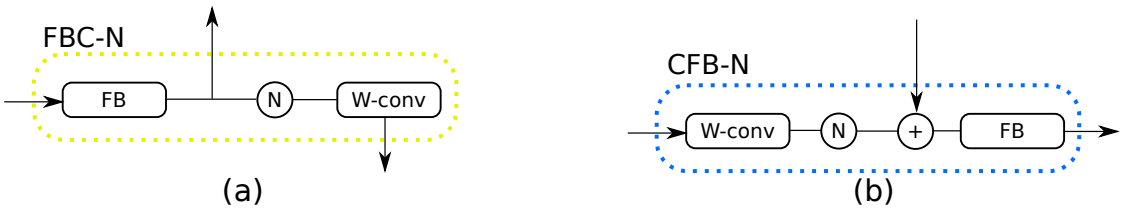

Fig. 3: a) FBC-N: (FB) Fourier Block, (C) W-Convolution and (N) scaling. Between the FB and N we get the features for the skip connection for the Decoder part. b) CFB-N: (C) W-Convolution, (FB) Fourier Block and (N) scaling. Between the N and FB we add the skip connection from the encoder part.

implemented in [29], however we differ in the use of the activation function. In the original work, they use a *ReLu* activation function in the FFC (they propose an in-painting method). However, recent works as [24] have proved that *ReLu* function is not really suited for depth estimation, since it is prone to make gradients vanish. Instead, we use *PReLu* as activation function, which is more stable for monocular depth estimation training [24]. The same activation function change has been made in the *Spectral block*[16] from the FB in order to homogenize the behavior of the network. The output of the FB is the information that we use as skip connection for the decoder part. After the FB, we re-scale the feature map by a factor of N, where N$<$1 means a down-scaling and N$>$1 an up-scaling, followed by a set of standard convolutions defined as W-conv. The W-conv block is defined as 3 consecutive Convolution-Batch normalization-Activation function blocks with circular padding, taking into account the continuity of panoramic images and their features. This W-conv block follows the ResNet[28] *bottleneck* structure for architecture homogeneity. As defined before, we use *PReLu* as activation function. The output of the W-conv is then added, and not concatenated, to the next scale-level output of the feature extractor, as seen in Fig. 2.

The decoder part follows the same principle as the encoder but in the inverse order, as shown in Fig 3b. The output of the encoder is fed to the decoder blocks (CFB-N) where the feature maps go through a W-conv block and then are up-scaled. The scaled features are added with the corresponding skip connection from the encoder, weighted with a learned parameter, and then go through a FB. The output of each decoder block is then fed to the next decoder block until we recover the full resolution of the network input.

B. Semantic segmentation branch

From the different feature maps generated by our encoder-decoder architecture, we need to extract the relevant information for the semantic segmentation task. We use the last five sets of feature maps from the decoder. We convolve and up-scale each set of features into an intermediate representation. However, instead of concatenating the feature maps, as done in previous works as [9], we make a learned-weighted sum of them to keep a more compact intermediate representation. Finally, we convolve this intermediate representation into the final representation of the semantic segmentation map. In this branch we switch to a *ReLu* activation function instead of the *PReLu* used in the main body of the architecture.

C. Depth estimation branch

For the depth estimation, we have created another separated branch that takes the last three blocks of feature maps from the decoder part. We propose to use these three sets of feature maps since they are the ones with higher resolution, then with higher level of details. As done in the semantic segmentation, we convolve and upscale the features to an intermediate representation, where we add them as a learned-weighted sum. Finally, we convolve again the intermediate representation to make the depth estimation. In this branch, we keep the *PReLu* activation function in the convolution from the feature maps to the intermediate representation. However, we switch to a *ReLu* function in the last convolution since depth cannot be a negative value. We tried different output functions, such as the *PReLu* activation function or not using any, but the *ReLu* function provided the best performance.

D. Loss functions

Semantic segmentation and depth estimation provide really different information of the environment. However these tasks have common characteristics that can benefit from each other, such as the objects boundaries [30]. We want to take advantage of these similarities making a joint training where the semantic segmentation and the depth estimation can be jointly predicted. For our training, we propose to train both branches, segmentation and depth estimation, at the same time and from the same input image. For the semantic segmentation loss L_{Seg}, we use the standard *Cross Entropy Loss* and weights for the classes [31], as a solution for the class imbalance in the dataset.

Similar to other state-of-the-art methods for monocular depth estimation [24][25], we use an *Adaptive Reverse Huber Loss* (B_c) as depth loss function L_{Dep}, defined as:

$$B_c(e) \begin{cases} |e| & |e| \leq c \\ \frac{e^2+c^2}{2c} & |e| > c \end{cases}, \tag{1}$$

where $e = Prediction - GroundTruth$ and c is defined as the 20% of the maximum absolute error for each training batch. Following the same idea as [24], we also define the loss function as the sum of the *Adaptive Reverse Huber Loss* on the depth map as well as the depth map gradients (approximated as Sobel Filters). The final L_{Dep} is computed as:

$$L_{Dep} = B_{c_1}(e) + B_{c_2}(\nabla_x) + B_{c_2}(\nabla_y), \tag{2}$$

where e defines the absolute depth error between the prediction and ground truth, ∇_x, ∇_y define absolute error between the x, y depth map gradients of the prediction and ground truth respectively, c_1 is the threshold in eq. 1 for the absolute depth map and c_2 is the threshold in eq. 1 for the gradients.

In addition to the semantic segmentation and depth estimation standard losses, we add another two losses to help in the joint training process. First, since the range of the depth estimation output should be greater than the semantic segmentation (i.e. the later is closer to a probability distribution while the former is a distance between 0 and, ideally, infinity), we add a term to force the depth estimation branch to fill the depth range between the closest and farthest pixels. To do so, we compute the mean square difference between the minimum and maximum values of prediction and ground truth in each batch as:

$$L_{mar} = \frac{\left(y_{gt}^{max} - y_{pred}^{max}\right)^2 + \left(y_{gt}^{min} - y_{pred}^{min}\right)^2}{2}, \quad (3)$$

where y_{gt}^{max}, y_{pred}^{max}, y_{gt}^{min} and y_{pred}^{min} are the maximum and minimum values of the ground truth and predicted depth maps respectively. Finally, to help in the object boundary definition, we propose to use an object oriented loss L_{obj}. This loss helps the network to better define the objects boundaries as well as create the depth discontinuities that appear in these boundaries. To compute the loss, we first compute per-class depth maps from the network depth prediction and semantic segmentation ground truth. Then we compute the mean of the *L1 Loss* of each class depth map to obtain the final L_{obj} as:

$$L_{obj} = \frac{1}{C}\sum_{i=0}^{C} L1({y_{gt}}^i, {y_{pred}}^i), \quad (4)$$

where C is the number of classes and ${y_{gt}}^i, {y_{pred}}^i$ are the ground truth and predicted class depth maps for the class i respectively.

Our final training loss is the combination of the previous losses. This joint loss function is computed as:

$$L_{total} = \alpha_1 \cdot L_{Seg} + \alpha_2 \cdot L_{Dep} + \alpha_3 \cdot L_{mar} + \alpha_4 \cdot L_{obj}, \quad (5)$$

where α_i are regularizers to weight the relevance of each individual loss in the final joint loss function. After a manual tuning of these parameters, we set these regularizers as $\alpha = [9.0, 14.0, 0.01, 5.0]$, obtaining the best performance for our network.

IV. Experiments

In this section we present a set of experiments to validate the joint training method proposed. We also make a comparison with a state-of-the-art architecture for depth estimation and semantic segmentation. Finally, we present qualitative results and application examples for our network.

For all the experiments we use the Stanford2D3DS dataset [32] with the first folder split, which uses *Area 5* as test set and the other areas for training and validation. We can only use the Stanford2D3DS dataset for both tasks since it is the only public dataset with semantic and depth information for real equirectangular panoramas. For our training, we perform data augmentations such as horizontal flipping, random horizontal rotation and color jitter.

TABLE I: Ablation study of Loss functions. Best validation metrics obtained on each training.

L_{Seg}	L_{Dep}	L_{mar}	L_{obj}	MRE	MAE	mIoU	mAcc
✓	✓	×	×	0.0613	0.0950	60.3	81.9
✓	✓	×	✓	0.0583	0.0855	61.5	82.5
✓	✓	✓	×	0.0560	0.0898	60.6	81.6
✓	✓	✓	✓	**0.0553**	**0.0827**	**62.7**	**84.2**
				Lower is better		*Higher is better*	

TABLE II: Ablation study of joint training. Best validation weights are used in each of the cases for the evaluation.

Training	MRE	MAE	mIoU	mAcc
Depth	0.1401	0.1773	-	-
Semantic	-	-	40.3	55.1
Joint	**0.0952**	**0.1327**	**46.1**	**63.1**
	Lower is better		*Higher is better*	

To evaluate our work and compare it with the state of the art, we use the following metrics. For the depth estimation task, we use the standard metrics introduced by [26]. We use the Mean Relative Error (*MRE*), Mean Absolute Error (*MAE*), Root-Mean Square Error of linear (*RMSE*) and logarithmic (*RMSElog*) measures, and three relative accuracy measures defined as the fraction of pixels where the relative error is within a threshold of 1.25^n for $n = 1,2,3$ ($\delta^1, \delta^2, \delta^3$). On the other hand, for the semantic segmentation task, we use standard metrics as the mean Intersection over Union (*mIoU*), computed as the average *IoU* for each class except the *Unknown* class; and the mean Accuracy (*mAcc*), computed as the average accuracy for each class except the *Unknown* class.

A. Ablation study

In this first subsection, we substantiate the joint training of depth and semantics and also evaluate the use of a combined loss function.

Loss Function. We evaluate how the overall performance of our network is affected by each loss function. For that purpose, we perform the same joint training using the task-specific losses and adding sequentially the new losses that we propose in Section III-D. We evaluate the performance of the network with four different metrics: two for depth estimation, *MRE* and *MAE*; and two for semantic segmentation, *mIoU* and *mAcc*. The results from Table I show that the use of these new losses increase the performance of our network. We have a greater improvement with the object oriented loss L_{obj} in the semantic segmentation. This loss uses depth and semantic information, providing an improvement in both branches as well as the shared encoder-decoder. In addition with the margin loss L_{mar}, which helps to improve the depth estimation, we observe that each task effectively benefit from each other.

Joint training. On a second step, we show that the joint training of depth estimation and semantic segmentation

TABLE III: Quantitative comparison for Depth Estimation and Semantic Segmentation on the Stanford 2D3DS dataset [32].

Network	MRE ↓	MAE ↓	RMSE ↓	RMSElog ↓	δ^1 ↑	δ^2 ↑	δ^3 ↑	mIoU ↑	mAcc ↑
HoHoNet[19]	0.1124	0.2265	0.4027	0.0710	0.8994	0.9687	0.9879	30.7	40.5
Ours	0.0952	0.1327	0.2727	0.0436	0.8424	0.9583	0.9863	46.1	63.1

benefits the overall performance of the network. For this purpose, we train our network for task-specific purposes, that means with only one of the branches at a time, and in the proposed joint training, with both branches. Notice that only half of the metrics are used for the task-specific training, since the other half correspond to a different task. In the task-specific training, we only use the specific loss function for each task, i.e. we only use L_{Dep} for the specific depth estimation training and L_{Seg} for the specific semantic segmentation training. We use the same metrics as in the previous experiment to compare the performance of the different training. The results presented on Table II confirm our first intuition, which confirm the results of [30]. The joint training of semantic segmentation and depth estimation increases the performance of the network for both tasks.

B. State of the art comparison

We compare our work with the sole state-of-the-art method which obtains semantic segmentation and depth estimation, HohoNet [19]. We train this network in the same conditions as our network and compare the performance of both proposals.

We train both networks in one GPU Nvidia RTX2080-Ti. The initial learning rate of our training is set to $1e-5$ and we use exponential decay and periodic step reductions in the learning rate. For HohoNet[19], we use their available code for training. The quantitative results of the evaluation are presented in Table III. We also present qualitative results and comparison of both networks in Figure 4.

An important remark is that our network provides, at the same time, depth and semantic information while HohoNet presents two slightly different architectures for task-specific solutions. That means, to obtain a depth map and a semantic segmentation with HohoNet we need to train two separate networks and make the inference of two networks. Using inference in parallel with HohoNet precludes the use of the optional depth input of the trained HohoNet network for semantic segmentation. For a fair comparison with our network, we use this scheme in our results providing only the RGB image as input information.

The quantitative comparison shows that our network has a better performance than HohoNet with the same input information. The results from HohoNet presented in this paper differ from the originals [19] since the training conditions are different. In this case, we only train both networks in the Stanford dataset, without pre-trained weights on other datasets for the whole network, using the *Area 5* only at test and not during the training or validation steps. In these conditions, FreDSNet performs better in the monocular depth estimation and semantic segmentation tasks.

C. Scene understanding for navigation

In this third subsection we present different results from our network and ideas of applications. With the combination of semantics and depth, we can extract the free space for navigation (extracting the floor) or compute where the obstacles are located (computing the position of the different segmented objects).

Navigation algorithms for mobile robots require to detect the obstacles and the free space around the vehicle. With an omnidirectional camera we can obtain RGB information of the surroundings in one shot. Then, FreDSNet can simultaneously obtain a semantic and depth maps of the environment. With this information, we can obtain different useful representations of the scene, allowing a better interaction of a mobile vehicle with the environment allowing a robot to be autonomous in unknown environments. Also, in the case of autonomous vehicles, it is important to be able to work in real time. We have evaluated the feasibility of our network for such a task obtaining that it can work at 33 frames per second with panoramas of 512×256 pixels of resolution (average speed computed with the test set of the dataset).

As an example of the information that can be obtained from our network, in Figure 5 we present several useful environment representations from a single equirectangular panorama. From left to right in the Figure, we show: free floor reconstruction, thought for terrestrial robot navigation; Room structure reconstruction, defines the maximum space that a vehicle can move; Room and obstacles, includes the room structure and the obstacles (in black) of the room; and the semantic reconstruction, which defines the environment and the different objects with which an autonomous vehicle can interact in the environment.

V. Conclusion

We have presented FreDSNet, a neural network for joint monocular depth estimation and semantic segmentation from single equirectangular panoramas. Our network is the first that exploits convolutions in the frequential domain for scene understanding. Also, we have proposed a joint training which improves the performance for both tasks, which has been validated experimentally on the Stanford2D3DS dataset.

The experiments performed validate our proposed contributions: the FFC is a good asset for indoor scene understanding allowing better understanding from more simple network architectures and the joint training mutually benefits the performance of both tasks. Besides, the comparison made shows that we provide slightly better results for both monocular depth estimation and semantic segmentation from equirectangular panoramas than the sole state-of-the-art network that also tackles both tasks under similar conditions.

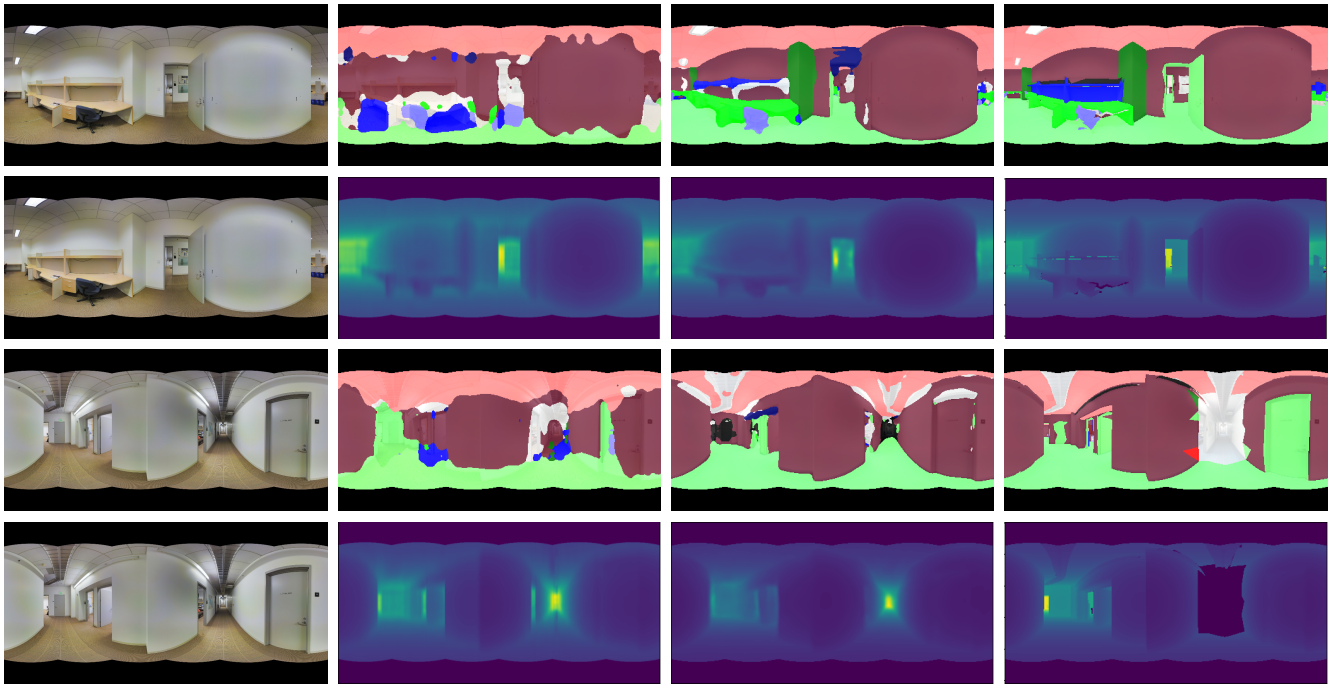

Fig. 4: Qualitative comparison between HohoNet [19] and our proposal for semantic segmentation and depth estimation in Stanford2D3DS [32].

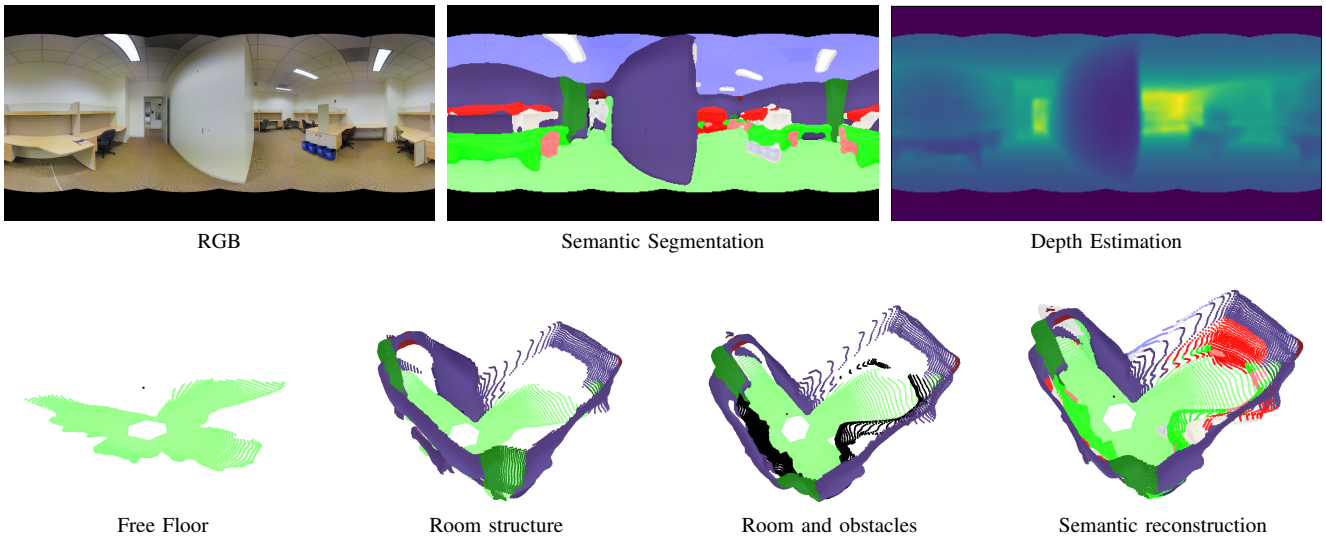

Fig. 5: In the first row: RGB is the input of our network which outputs the Semantic Segmentation and Depth estimation. In the second row: different useful environment representations that can be obtained from the output information provided by FreDSNet. (For a better representation, the ceiling has been removed from all visualizations)

The research of scene understanding methods is still an open topic. Many different approaches are appearing, making improvements in key aspects. In this work we provide a novel solution which information can be used in many others research fields such as virtual or augmented reality, robot navigation and interaction with the environment.

ACKNOWLEDGEMENT

This work was supported by projects PID2021-125209OB-I00 (MCIN/ AEI/ 10.13039/ 501100011033 and FEDER/UE), TED2021-129410B-I00 (MCIN/ AEI/ 10.13039/ 501100011033 and NextGenerationEU/PRTR) and JIUZ-2021-TEC-01.

REFERENCES

[1] M. Naseer, S. Khan, and F. Porikli, "Indoor scene understanding in 2.5/3d for autonomous agents: A survey," *IEEE Access*, pp. 1859–1887, 2018.

[2] C. Zou, J.-W. Su, C.-H. Peng, A. Colburn, Q. Shan, P. Wonka, H.-K. Chu, and D. Hoiem, "Manhattan room layout reconstruction from a single 360 image: A comparative study of state-of-the-art methods," *International Journal of Computer Vision*, pp. 1410–1431, 2021.

[3] B. Berenguel-Baeta, J. Bermudez-Cameo, and J. J. Guerrero, "Scaled 360 layouts: Revisiting non-central panoramas," in *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops*, pp. 3702–3705, IEEE, 2021.

[4] C. Fernandez-Labrador, J. M. Facil, A. Perez-Yus, C. Demonceaux, J. Civera, and J. J. Guerrero, "Corners for layout: End-to-end layout recovery from 360 images," *Robotics and Automation Letters*, pp. 1255–1262, 2020.

[5] G. Pintore, M. Agus, and E. Gobbetti, "Atlantanet: Inferring the 3d indoor layout from a single 360 image beyond the manhattan world assumption," in *European Conference on Computer Vision*, pp. 432–448, Springer, 2020.

[6] C. Sun, C.-W. Hsiao, M. Sun, and H.-T. Chen, "Horizonnet: Learning room layout with 1d representation and pano stretch data augmentation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1047–1056, IEEE, 2019.

[7] I. Rusli, B. R. Trilaksono, and W. Adiprawita, "Roomslam: Simultaneous localization and mapping with objects and indoor layout structure," *IEEE Access*, pp. 196992–197004, 2020.

[8] M. Salas, W. Hussain, A. Concha, L. Montano, J. Civera, and J. Montiel, "Layout aware visual tracking and mapping," in *International Conference on Intelligent Robots and Sistems*, pp. 149–156, IEEE, 2015.

[9] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid, "Blitznet: A real-time deep network for scene understanding," in *Proceedings of the International Conference on Computer Vision*, pp. 4154–4162, 2017.

[10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the International Conference on Computer Vision*, pp. 2961–2969, IEEE, 2017.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, pp. 211–252, 2015.

[12] M. Eder, M. Shvets, J. Lim, and J.-M. Frahm, "Tangent images for mitigating spherical distortion," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 12426–12434, IEEE, 2020.

[13] J. Guerrero-Viu, C. Fernandez-Labrador, C. Demonceaux, and J. J. Guerrero, "What's in my room? object recognition on indoor panoramic images," in *International Conference on Robotics and Automation*, pp. 567–573, IEEE, 2020.

[14] H. Koppula, A. Anand, T. Joachims, and A. Saxena, "Semantic labeling of 3d point clouds for indoor scenes," *Conference on Neural Information Processing Systems*, 2011.

[15] X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3d recurrent neural networks with context fusion for point cloud semantic segmentation," in *European Conference on Computer Vision*, Springer, 2018.

[16] L. Chi, B. Jiang, and Y. Mu, "Fast fourier convolution," in *Conference on Neural Information Processing Systems*, pp. 4479–4488, 2020.

[17] S. Zheng, M.-M. Cheng, J. Warrell, P. Sturgess, V. Vineet, C. Rother, and P. H. S. Torr, "Dense semantic image segmentation with objects and attributes," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2014.

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, 2014.

[19] C. Sun, M. Sun, and H.-T. Chen, "Hohonet: 360 indoor holistic understanding with latent horizontal features," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 2573–2582, IEEE, 2021.

[20] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera, "Cam-convs: Camera-aware multi-scale convolutions for single-view depth," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 11826–11835, IEEE, 2019.

[21] M. Heo, J. Lee, K.-R. Kim, H.-U. Kim, and C.-S. Kim, "Monocular depth estimation using whole strip masking and reliability-based refinement," in *European Conference on Computer Vision*, pp. 36–51, Springer, 2018.

[22] J.-H. Lee and C.-S. Kim, "Monocular depth estimation using relative depth maps," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, IEEE, 2019.

[23] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun, "Dense monocular depth estimation in complex dynamic scenes," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, IEEE, 2016.

[24] G. Pintore, M. Agus, E. Almansa, J. Schneider, and E. Gobbetti, "Slicenet: deep dense depth estimation from a single indoor panorama using a slice-based representation," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 11536–11545, IEEE, 2021.

[25] F.-E. Wang, Y.-H. Yeh, M. Sun, W.-C. Chiu, and Y.-H. Tsai, "Bifuse: Monocular 360 depth estimation via bi-projection fusion," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 462–471, IEEE, 2020.

[26] N. Zioulis, A. Karakottas, D. Zarpalas, and P. Daras, "Omnidepth: Dense depth estimation for indoors spherical panoramas," in *European Conference on Computer Vision*, pp. 448–465, Springer, 2018.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 770–778, IEEE, 2016.

[29] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *Proceedings on the Winter Conference of Applications of Computer Vision*, pp. 2149–2159, IEEE, 2022.

[30] Z. Zhang, Z. Cui, C. Xu, Z. Jie, X. Li, and J. Yang, "Joint task-recursive learning for semantic segmentation and depth estimation," in *European Conference on Computer Vision*, pp. 235–251, Springer, 2018.

[31] J. Tian, N. C. Mithun, Z. Seymour, H.-P. Chiu, and Z. Kira, "Striking the right balance: Recall loss for semantic segmentation," in *International Conference on Robotics and Automation*, IEEE, 2022.

[32] I. Armeni, S. Sax, A. R. Zamir, and S. Savarese, "Joint 2d-3d-semantic data for indoor scene understanding," *arXiv preprint arXiv:1702.01105*, 2017.