Get the Best Out of Depth and Surface Normals by Integrating Them

Mehran Aghabozorgi* — Simon Fraser University
Mahdieh Ghane* — Simon Fraser University
Mahdi Miangoleh — Simon Fraser University
Yağız Aksoy — Simon Fraser University

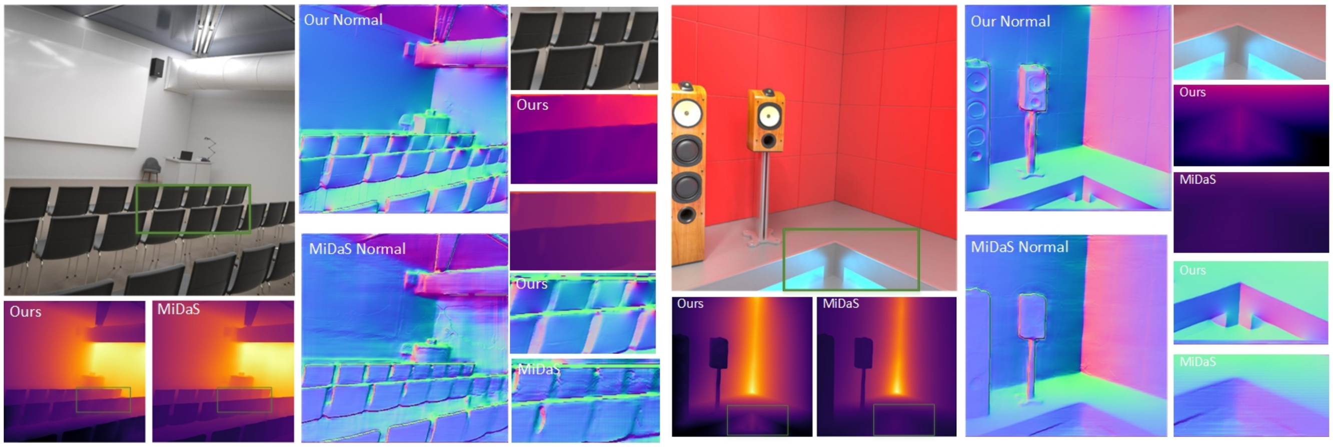

Figure 1: Two examples of applying our method on depth maps produced by MiDaS [Ranftl et al. 2020]. Normals are generated using the corresponding depth maps as explained in Sec 2.1. Our improved depth map contains more details and better represents flat surfaces.

ACM Reference Format:
Mehran Aghabozorgi, Mahdieh Ghane, Mahdi Miangoleh, and Yağız Aksoy. 2025. Get the Best Out of Depth and Surface Normals by Integrating Them. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In recent years, data-driven approaches [Eigen et al. 2014; Fu et al. 2018; Godard et al. 2016, 2019; Miangoleh et al. 2021; Ranftl et al. 2020] have shown great progress in estimating depth and normal maps from a single scene. Despite their impressive results, we often see some inconsistencies between these depth and normal estimations. For example, depth maps tend to have poor representation of flat surfaces, and fine-grained details while normal estimations usually do better on flat surfaces.

In this work, we investigate the possibility of integrating depth maps coming from [Ranftl et al. 2020] and normal maps from [Kar et al. 2022] to get the best of both by improving the resulting depth estimation. We formulate this as a blending problem that tries to integrate a pair of depth and normal maps to generate an improved depth map. We try to keep our estimated and input depth maps consistent with each other while using the information available in both of our input maps to get rid of their drawbacks. We describe our method detail in Sec 1.

2 FORMULATION

In this section, we describe our problem formulation by motivating and connecting our approach with the well-known Poisson blending problem setup.

2.1 Depth and Normal Relation

As described in [Antensteiner et al. 2018], given a depth map, we can convert it to its corresponding normal map by using the Sobel [Kanopoulos et al. 1988] operator and normalizing the results, i.e., $N_x = \frac{-\nabla_x Z}{|(\nabla Z, 1)|_2}$, $N_y = \frac{-\nabla_y Z}{|(\nabla Z, 1)|_2}$ and $N_z = \frac{1}{|(\nabla Z, 1)|_2}$ where Z is the depth map and $N(Z) = (N_x, N_y, N_z)$ is its corresponding normal vector. However, there is no straightforward way to get from the normal map to depth. As described above, normals contain gradient information of depth. This incentivizes a gradient-domain formulation to make use of this available information in the normal map.

2.2 Consistent Depth and Normal Scales

Depth maps generated by MiDaS [Ranftl et al. 2020] are different from the ground-truth depth by a scale and bias factors. This makes it harder to integrate our input depth and normal maps with each other as the scales are different. To tackle this, we first solve for *a*

*Denotes equal contribution.

Conference'17, July 2017, Washington, DC, USA
© 2025 Association for Computing Machinery.
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

and b such that $min_{a,b}\left[(N(aZ_{MiDaS}+b)-N_{Omnidata})^2\right]$ where $N(.)$ is defined above, Z_{MiDaS} is the depth map generated by MiDaS and $N_{Omnidata}$ is the normal map generated by Omnidata [Kar et al. 2022]. We use a standard least square solver to solve this. After applying a and b, the depth map's gradients are more consistent with the input normal map. This sets us up to describe our blending formulation below.

Our formulation is inspired by Poisson blending [Martino et al. 2016]. Given a source, a target, and a mask, Poisson blending makes use of the gradient information to seamlessly blend a portion of the source defined by the mask into the target image. Formally, Poisson formulation for a 1D source and target can be defined as:

$$v = \underset{v}{\operatorname{argmin}} \sum_{i\in M} \sum_{j\in N_i\cap M} \left((v_i - v_j) - (s_i - s_j)\right)^2 + \sum_{i\in M} \sum_{j\in N_i\cap \sim M} \left((v_i - t_j) - (s_i - s_j)\right)^2 \quad (1)$$

where s and t are source and targets respectively, M is a $0-1$ hard-mask and N_i is the neighbouring pixels of pixel i. An example of such blending can be seen in Fig 2.

This formulation is similar to what we want, however, we want to be able to use the information available in both depth and normal maps and control their relative importance in each pixel based on their reliability in that region. This suggests a soft-mask formulation instead of the hard-mask one defined above.

Specifically, each pixel in our mask is a value between 0 and 1 where 0 means the information completely comes from the depth map intensity, 1 means it completely comes from the normal map intensity (which contains gradient information), and in between values mean it comes both from the normal map intensities and depth map *gradients*. The amount by which we use either of these values is controlled by the mask value itself.

As described above, for in-between values, we make use of the depth map's gradients instead of its intensities directly. We do this because we observe that in some cases, fitting a and b is not enough to make the scale of depth intensities and normal intensities consistent while the gradient scales are more consistent. This enables us to better make use of the depth map information. Formally, our formulation can be defined as:

$$v = \underset{v}{\operatorname{argmin}} \sum_{i\in M} \sum_{j\in N_i\cap M>0} \Big((v_i - v_j) - \left[M_i\,(N_i) + (1-M_i)\,(Z_i - Z_j)\right]\Big)^2 + \sum_{i\in M} \sum_{j\in N_i\cap M=0} \left((v_i - Z_j) - (N_i)\right)^2 \quad (2)$$

where Z is the input depth map, N is the input normal map after applying a and b to match the scales, and M is our soft-mask. Note that, N_i already contains the gradient information in itself (Sec 2.1), that's why we don't need $N_i - N_j$ above. For the sake of simplicity, we have presented our formulation in 1D, it can readily be extended to 2D.

2.3 How to Choose the Mask?

Depth and normal maps each have their own specific pros and cons. Our goal is to design our soft-mask to use the part of information from each that are more reliable. For example, we know that normal map are not reliable on edges. Moreover, we observe that depth map estimations from MiDaS struggle to represent flat surfaces (Fig 4). These observations led us to define our mask as follow:

- For strong edges, defined by a threshold, we set our mask to 0. This means the information comes from the depth intensities.
- For less strong edges, we set the mask to a value between 0 and 1 (getting information from both normal intensities and depth gradients) depending on the intensity of the edge. The stronger the edge, the closer the value of the mask to 0.
- We randomely select a few patches and set the mask to a value close to 1, which is analogous to getting most of the information from the normal intensities and a bit from depth map gradients.
- Other parts of the mask are by default set to 1 which means the information solely comes from the normal intensities. These sections mostly represent flat surfaces.

These choices are based on our observations about where each depth and normal maps are more useful. Finally, we erode (using imerode in Matlab) our mask to further avoid using edges information from the normal intensities. In Fig 3 we show that not eroding will result in getting unreliable information from the normal map!

3 RESULTS AND EVALUATION

In Fig 4 we present some of our estimated depth maps, their corresponding normal maps and the soft-mask used to produce them. We can see that our results better represent details and flat surfaces. We further use Depth Discontinuity Disagreement Ratio (D^{3}R) proposed in [Miangoleh et al. 2021] to evaluate the quality of high frequency details in depth estimations. As it can be seen in Table 1, our results achieves lower D^{3}R error in comparison to those of MiDaS. We also have provided some 3D results here.

Table 1: D^{3}R comparison of our results and MiDaS. Lower numbers are better.

	Speaker	Room 3605	Room 7600
MiDaS [Ranftl et al. 2020]	0.243	0.067	0.106
Ours	**0.189**	**0.053**	**0.092**

4 CONCLUSION

We have demonstrated that it is possible to make use of both depth and normal maps in order to improve the depth estimation. Our simple formulation is fast and applicable to high resolution inputs. As a future extension, an easy to use interface can be designed so that the user can easily use our method while being able to control the method's parameters (such as edge threshold). Moreover, our method can be used in downstream data-driven approaches to improve the results.

REFERENCES

D. Antensteiner, S. Stolc, and Thomas Pock. 2018. A Review of Depth and Normal Fusion Algorithms. *Sensors* 18, 2 (2018). https://doi.org/10.3390/s18020431

David Eigen, Christian Puhrsch, and Rob Fergus. 2014. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger (Eds.), Vol. 27. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2014/file/7bccfde7714a1ebadf06c5f4cea752c1-Paper.pdf

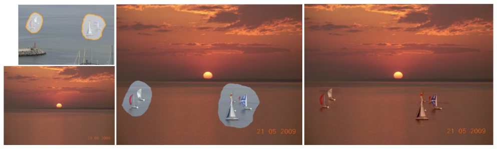

Figure 2: A Poisson blending [Martino et al. 2016] example. It uses the gradient information in the source to seamlessly blend it into the target.

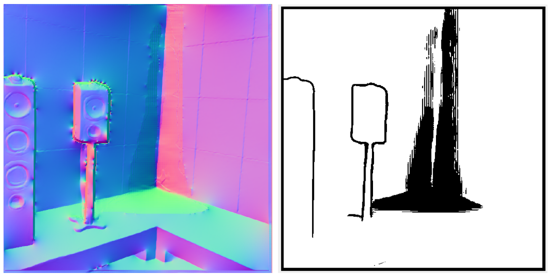

Figure 3: An example of the case where we use hard-mask without eroding. The resulting image is flawed because 1- we are using edge information from the normal and 2- we are using a hard-mask and the scale of depth intensities and the input normal map are not matched.

Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. 2018. Deep Ordinal Regression Network for Monocular Depth Estimation. (06 2018).

Clément Godard, Oisin Aodha, and Gabriel Brostow. 2016. Unsupervised Monocular Depth Estimation with Left-Right Consistency. (09 2016).

Clément Godard, Oisin Aodha, Michael Firman, and Gabriel Brostow. 2019. Digging Into Self-Supervised Monocular Depth Estimation. https://doi.org/10.1109/ICCV.2019.00393

N. Kanopoulos, N. Vasanthavada, and R.L. Baker. 1988. Design of an image edge detection filter using the Sobel operator. *IEEE Journal of Solid-State Circuits* 23, 2 (1988), 358–367. https://doi.org/10.1109/4.996

Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 2022. 3D Common Corruptions and Data Augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 18963–18974.

J. Martino, Gabriele Facciolo, and Eva Llopis. 2016. Poisson Image Editing. *Image Processing On Line* 5 (11 2016), 300–325. https://doi.org/10.5201/ipol.2016.163

S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yagiz Aksoy. 2021. Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 9680–9689.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. 2020. Towards Robust Monocular Depth Estimation: Mixing Datasets for Zero-shot Cross-dataset Transfer. *IEEE Transactions on Pattern Recognition and Machine Intelligence* (2020).

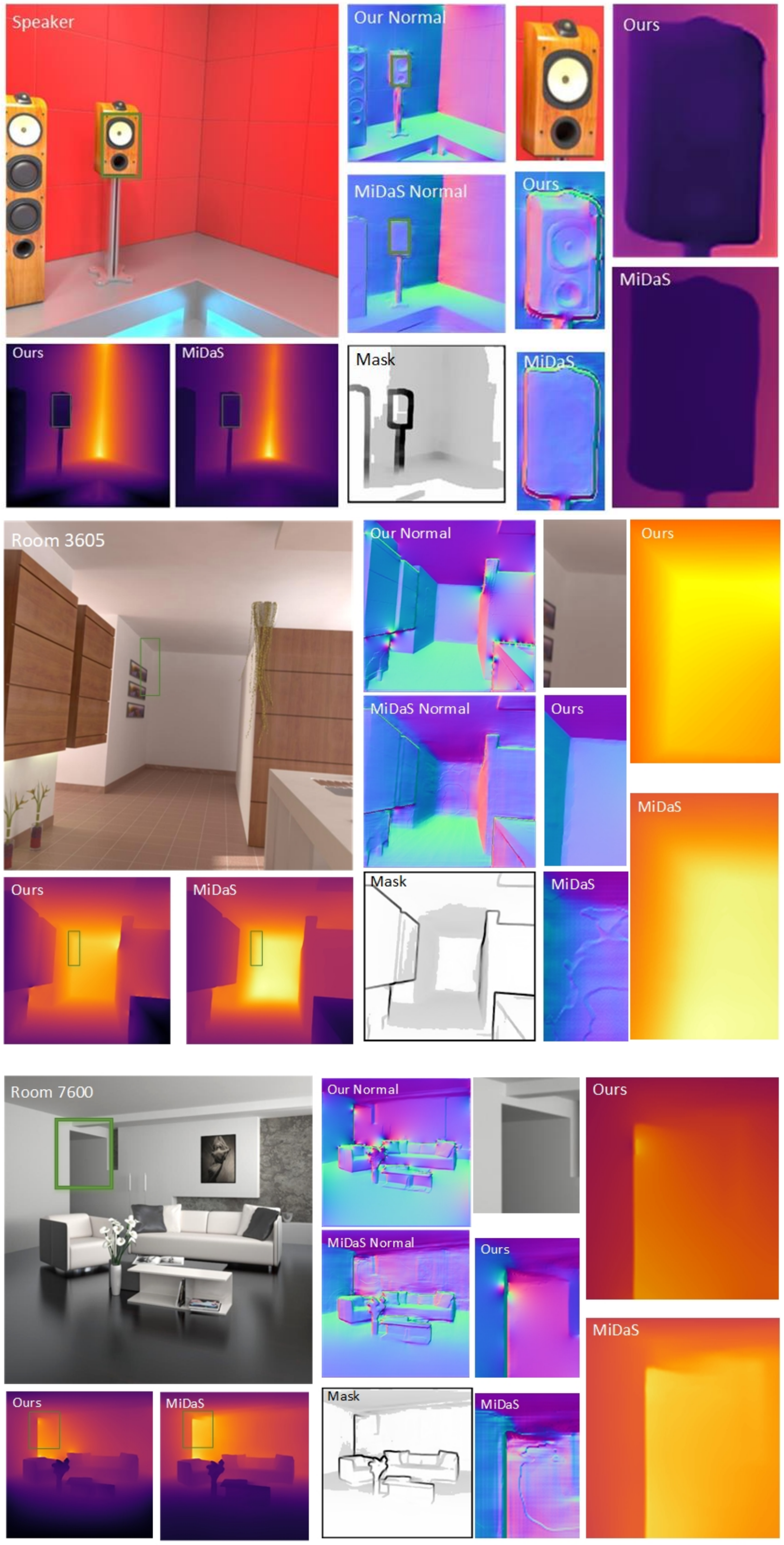

Figure 4: Here we demonstrate the qualitative results of the samples reported in Table 1. For each example, we can see Our generated depth, MiDaS depth and their corresponding normals produced using the same method outlined in Sec 2.1. We also show the mask that was used to generate our depth.