Piecewise Monocular Depth Estimation by Plane Fitting

Jan Coccejus Jan Helge Dörsam Korbinian Kunst Felix Wirth
Technical University of Darmstadt
https://github.com/jhd-old/Deep-Learning-Lab

Abstract

This paper proposes a modified approach for dense depth estimation from monocular images. We model a complex 3D scene via over-segmentation with superpixels as a piecewise planar and rigid approximation. Based on this assumption we represent every planar structure by plane coefficients (i.e. unnormalized normal vectors). In this way, we solve the homogeneous depth estimation problem that our baseline architecture Monodepth2 from Godard et al. *2019 suffered. In particular, we propose (i) a Normal-2-Depth block inside the architecture that estimates surface normal coefficients, (ii) a Superpixel Loss that incorporates superpixel information and exploits sharper edges and (iii) a Normal Loss that ensures homogeneous depth for planar surfaces. We demonstrate the effectiveness of the proposed improvements in a detailed depth map analysis and show comparable scoring metrics with state-of-the-art results on the KITTI Eigen split test set.*

1. Introduction

We aim to estimate dense depth maps from single monocular images. This approach is fundamentally an ill-posed problem because a single 2D view of a scene can be created by many 3D scenes, which can be explained, inter alia, due to scale ambiguity. In the classical computer vision, this has been solved with stereo-vision and epipolar geometry. In the era of deep neural networks, many proposals have been made to solve this issue with CNN's. Knowing depth information is important for domains from robotic surgery [54] to urban automated driving [48] and volume rendering from two images [26].

Our method is built on top of one of the state-of-the-arts method, benchmarked on the KITTI Eigen [9] test set, Monodepth2 from Godard *et al*. [16]. Though it has to be said that the created depth maps still have their shortcomings. In particular non-lambertian surfaces cause deviation from the physical world, e.g. windows and car windshields are not represented with homogeneous depth. Further the edges of objects tend to fade out and lack abrupt depth change between different objects. Motivated by these two observations, this work enforces using over-segmentation of a given scene via superpixels and creating a piecewise planar scene model. It is then assumed that each of these planes can be represented by normals perpendicular to this plane. An illustration of the result of this method is shown in figure 1.

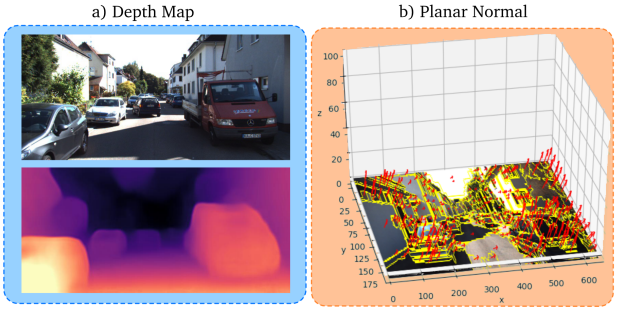

Figure 1. **Results from a single image**. **(a) Depth map:** Our depth prediction results on a randomly picked image from KITTI [13]. **(b) Planar unnormalized normal vectors:** To the chosen image the according over-segmented image with surface normals as produced by our network are plotted beginning in the center of mass for each planar structure. The magnitude has been scaled due to visibility reasons.

In this work we present the following contributions:

- We introduce an approach to improve the depth estimation for homogeneous surfaces by leveraging over-segmentation of images with superpixel in the loss-function.
- We present a Normal-2-Depth block inside the network that successfully outputs plane coefficients (i.e. unnormalized normal vectors). The experiment shows that 3D normal vector maps can be calculated from these coefficients.
- We propose a novel loss function that implements superpixel and normal information to establish sharp edges and homogeneous depth estimations for planar surfaces.

2. Related Work

We review Neural Networks that take a single color image as input at test time and predict the depth of each pixel as an input (dense depth estimation). Further approaches that incorporate superpixel over-segmentation and surface normals for depth estimation, even though not implemented in neural network architectures yet, are summarized.

2.1. Monocular Depth Estimation

Monocular depth estimation deals with the problem of estimating depth from single images, whose concept is fundamentally different from using stereo pairs because it is not possible to triangulate points and make use of epipolar geometry. In a monocular setting contextual information is required, e.g. texture variation, occlusions, and scene context [17]. The depth information cannot reliably be calculated from local image patches. Consider for example a patch of brown pixels which could either represent a distant tree or a chair in the foreground. The challenge is to make use of global information of the picture to correctly determine depth by modeling context information with manually engineered features like done by Saxena *et al*. in Make3D [38] or since the era of NN by using CNNs.

Eigen *et al*. presented an architecture in 2014 [10] that successfully estimated depth in a supervised manner using LIDAR ground truth and a CNN to minimize a scale invariant mean squared error. The network itself consists of two deep networks to estimate global and local structures that refine finer resolutions. Using depth ground truth from LIDAR suffers a few problems though. First of all it sparse and the results might be inaccurate due to infrared interference or to surface specularities. Further, it is expensive and time-consuming to generate. This motivated unsupervised learning methods because they apply great practical use for complex scenes where ground truth data is hard to obtain. In particular self-supervised learning has the following benefits: a) training does not need labeled data, (b) dense depth map reconstruction if prior knowledge can be automatically incorporated, and (c) leveraging multiple interconnected tasks in classical structure from motion like optical flow, and camera intrinsic estimation [5]. For this, the depth map generation is phrased as a novel-view synthesis problem for most modern monocular depth estimation architectures.

Xie *et al*. [49] address this problem by using depth image-based rendering algorithms for usage in NN incorporating the idea to generate the corresponding right view from an input left image of a pair of binocular images. A network based on VGG16 [40] predicts a probabilistic disparity map that is then used by a differentiable depth image-based rendering layer to produce the right view. In a second approach of self-supervised depth estimation Godard *et al*. propose Monodepth [15] where training data consists of pairs of left and right images. A disparity prediction model can be trained to warp the left image into the right image using photometric reconstruction losses. Depth can be recovered from the disparities by using the camera intrinsics making depth ground truth data obsolete at train time. While stereo pairs are necessary during training, during test time depth can be predicted from a single image. Zhou *et al*. faced the problem differently in the same year with adopting a depth and pose-net to estimate depth and camera motion at the same time by using a differentiable depth-image-base renderer [28][58]. The loss measures the photometric difference between the synthesized and the actual image, where synthesis is obtained by a 3D reconstruction in the first frame with back-projected pixel intensities, followed by rigid displacement and perspective projection in the second frame.

Cheng *et al*. [5] continue the approach of using a separate depth and camera net but additionally add a flow net to create a geometrically-inspired learning framework to jointly learn depth, flow and camera pose. While continuing the novel-view synthesis proposed by Zhou *et al*. [58] they modify the loss and propose an adaptive photometric loss that represents the minimum photometric error between the displacements characterized by optical flow or rigid motion. The similarity measure between two pixels stays the same and is weighted by the SSIM and L1 components. Further advances include enforcement of 3D structural consistency and epipolar constraint for the optical flow determination. Godard *et al*. [16] also make use of the architecture of Zhou [58] and implement the photometric loss ideas of 2017 in this architecture while also adding a minimum reprojection loss to handle occlusions and an auto masking loss. Besides the shortcomings of LIDAR a survey of current research papers [10] [12] [49] [45] [15] [58] [16] [5] shows, that all methods are benchmarked on the KITTI dataset [14] [13] which uses LIDAR for the data collection and provides the ground truth. Further datasets with laser ground truth like 3DMake [38] from 2009 and NYUD-V2 [27] from 2012 exist. Another technique to use depth ground truth is to create synthetic images like in the FlyingThings dataset [49]. Due to majority accordance with the community, we will also train and evaluate our approach on the KITTI dataset and the responding splits (e.g. Eigen [9] and Zhou split [58].)

2.2. Depth Estimation by Exploiting Superpixel Relations and Surface Normals

Although recent methods have achieved impressive progress in monocular depth estimation they neglect geometric constraints for the 3D space. Geometric constraints such as plane coefficients and plane normal vectors can be used to improve existing approaches. Depth and surface normal vectors are highly correlated information. Due to this correlation surface normals (or plane coefficients) can

be useful information to understand 3D scenes.
Eigen *et al*. [9] proposed a multi-scale convolutional network that outputs depth, surface normals and semantic labels. In contrast to the effort of this work to first model the complex 3D scene via normals, Eigen *et al*. predicted depth and then calculated back to normals via a method proposed by Silberman [27]. This idea was further developed in GeoNet by Qi *et al*. [32] who directly estimated normals with two CNNs but learned them based on ground truth information of the NYU-V2 dataset and pre-train the depth-to-normal network taking ground truth depth as input [27].

Ying *et al*. used a modern Network Structure MobileNetV2 [37] to calculate Virtual Normals (VN) [19]. The network learns via a loss term that enforces that the VN directions, which are calculated by randomly sampled three points in the reconstructed 3D space, follow a geometric constraint, which is defined as angular difference between ground truth and calculated surface normals. The training itself is done with ground truth information from the NYU-V2 dataset [27] [55].

Learning normals via ground truth is also used by Yang *et al*. [50]. Therefore depth-normal consistency [51] is combined with novel-view synthesis by Zhou [58] with an edge estimation network that is pretrained on semantic segmentation ground truth with the CityScapes dataset [6].
Besides exploiting normal geometric constraints it can be useful to approximate surfaces as superpixel. Superpixel in general aim to group perceptually similar pixels and were first introduced by Ren and Malik in 2003 [34]. Superpixel have become an established image preprocessing step to significantly reduce the complexity of higher-level computer vision techniques [53]. Today there is a variety of different superpixel methods with different approaches. Stutz *et al*. break down the different superpixel methods according to their approaches and compare them using established metrics [42][43]. Further researchers have also classified and evaluated the superpixel methods [46][41][2][25][3][29]. Superpixel have been used in Neural Networks as superpixel pooling layer for semantic segmentation [39][44][22].
In addition to solve the relative depth ambiguity via CNNs, structure from motion (SfM) techniques have been developed and in particular superpixels have been incorporated in these approaches [20] [21] [8]. Kumar *et al*. propose a "Superpixel Soup" algorithm to reconstruct a dense 3D model for a complex dynamic and non-rigid scene with images taken by a moving monocular camera [20]. By approximating a dynamic scene as piecewise planar and rigid they solve the relative scale ambiguity in structure-from-motion. In particular, they have the assumption i) that the transformation between two frames is locally piecewise rigid and globally as rigid as possible (ARAP). Meaning a deformation is not arbitrary but regular regarding rigidity. Second ii) a 3D scene can be approximated by a piecewise planar structure with the properties of being piecewise smooth for the following frames. Based on these assumptions for a given pair of two monocular image frames and dense optical flow correspondences the images are over segmented into 1000-2000 superpixels via the SLIC algorithm [1] and construct a K-NN graph to represent the given scene as a graph defined over superpixels. To recover the rigid motion and 3D geometry for each 3D superpixel the two-view epipolar geometry is used and finally the proposed energy function is optimized to align and assemble the reconstructed superpixels. This energy function is non-convex and solving it takes 12 min for a 1024x436 pixel image on a regular desktop computer. Though a direct application to a neural network architecture is not possible. The same authors published a follow-up work that omits explicit estimation of 3D translation and rotation using 1000-1200 superpixels [21]. Di *et al*. [8] also solve the inherent relative scale ambiguity (RSA) problem that exists when depth needs to be estimated by exploiting neighboring superpixel relations. In particular, the motion relations between to consecutive frame are used to predict spatial relations. Here also the assumption of a piecewise planar model is made. The preferred superpixel algorithm is again SLIC with a fixed size of 150 pixels per superpixel.

2.3. Monodepth2 as Baseline Architecture

In this work, we utilize the architecture of Godard *et al*. (2019) [16] as a baseline, which is a follow-up to their work from 2017 [15], that uses temporal sequences of frames to be trained in a self-supervised setting on stereo pairs images or with successive monocular frames, where depth (disparities) are an intermediate result in a right-to-left image reconstruction pipeline. The core idea is to generate a differentiable warping between two images and using the photometric difference (SSIM [47]) as supervision. For the quality of the image reconstruction and disparity prediction, a combination of different losses is used, including L1-image-reconstruction loss and left-right disparity smoothness.
In Monodepth2 the novel view synthesis method of Zhou *et al*. [58] is adopted and the left-right consistency dropped. Instead a minimum reprojection loss to handle occlusions , a full resolution multi-scale network to reduce artifacts and a auto-masking loss to ignore pixels that violate camera motion assumptions are proposed [16]. One potential shortcoming in this approach is that these losses are local pixel-wise, while the approaches made in this work enforce more pixel-neighborhood consistency. The architecture itself has been deeply studied by the authors and is visualized and explained in the complementary material, see figure 9.

3. Methodology

Our proposed architecture combines the idea of superpixels and geometric constraints in the monocular depth estimation pipeline. In the following we introduce the implementation of our approaches which are summarized in figure 2.

3.1. Normal-2-Depth Block

Throughout the text we also refer to the plane coefficient as (unnormalized) surface normal. Our baseline's depth net [16] outputs one depth value per pixel. We have modified the net similar to Eigen *et al*. [9] so that it outputs three coefficients a, b, c instead of one depth value.

Since the KITTI dataset [14] contains only depth from LIDAR as ground truth and no information about vector normals, the vector normals must be calculated back to depth. We created the Normal-2-Depth block to calculate the per-pixel depth map from the three coefficients using the pinhole camera model. The relationship between the normal vector and a point in 3D space can be described as follows:

$$ax + by + cz = n^T \cdot P = 1, \tag{1}$$

where a, b, c are the plane coefficients (i.e., unnormalized normal vectors), n^T is a vector of the coefficients, and x, y, z, respectively P describe the point in 3D space. Using the pinhole camera model from which the equations 2 and 3 are derived, we can now derive equation 4. The point in 2D space is described by p and K is the by the KITTI dataset provided intrinsic camera matrix.

$$p = K \cdot P. \tag{2}$$

$$P = (K^{-1}p) \cdot z. \tag{3}$$

$$z = d = \frac{1}{n^T \cdot (K^{-1}p)}. \tag{4}$$

Since the selected network also expects a disparity map, we calculate it directly from the depth map using the following relation:

$$disp = \frac{f \cdot b}{d}, \tag{5}$$

where $disp$ is the per-pixel disparity map, f is the focal length, b is the used distance between the stereo cameras from the KITTI dataset and d is the per-pixel depth map.

3.2. Superpixel Input

SLIC [1] is the preferred algorithm for superpixels in depth estimation, see section 2.2. Since SLIC divides the image into equally sized superpixels, the question arises whether equally sized superpixels can approximate planes in the image scene well. For the optimal coverage of the areas to be approximated, we therefore first use a superpixel algorithm, which merges contiguous pixels into a superpixel area and thus generates superpixels of different sizes that are optimally adapted to the area. An algorithm of this kind is Felzenswalb's method [31]. For a deeper understanding of the influence of the type of superpixel we compare the results of Felzenswalb's method additionally with SLIC. The superpixels are calculated offline for each image of the KITTI Eigen-Zhou split. Each superpixel region is labeled by a unique integer which leads to single-channel data.

To the best of our knowledge superpixel information can be used in different ways:

- Additional to the image as input for the encoder.
- As a superpixel pooling layer.
- In the loss function.

In our proposed framework we implemented all of these three ways to use superpixel information. First of all the baseline's encoder is modified to be able to retrieve inputs with different channel sizes:

- 3 channel input: RGB image.
- 4 channel input: RGB image and superpixel data.
- 6 channel input: RGB image and RGB image averaged over each superpixel.

The superpixel pooling and the superpixel information are used in the Superpixel Loss and in the Normal Loss.

3.3. Superpixel Loss

The Monodepth2 loss is constructed by two losses. The minimum reprojection loss L_p measures similarity of a source and a target image, which is warped using depth and relative camera pose information estimated by the network. The smoothness loss ensures sharp edges and smooth disparity gradients on surfaces. For a detailed explanation, see supplementary material 7.2. The Monodepth2 loss is defined as:

$$\begin{aligned} L &= \mu L_p + \lambda L_s \\ &= \mu min(\frac{\alpha}{2}(1 - SSIM(I_a, I_b)) + (1-\alpha)\left\|I_a - I_b\right\|_1) \\ &+ \lambda(|\delta_x d_t|\, e^{-|\delta I_x|} + |\delta_y d_t|\, e^{-|\delta I_y|}) \end{aligned} \tag{6}$$

Zhao *et al*. [57] evaluated, that a combination of an SSIM and l_1-norm based loss is beneficial for image

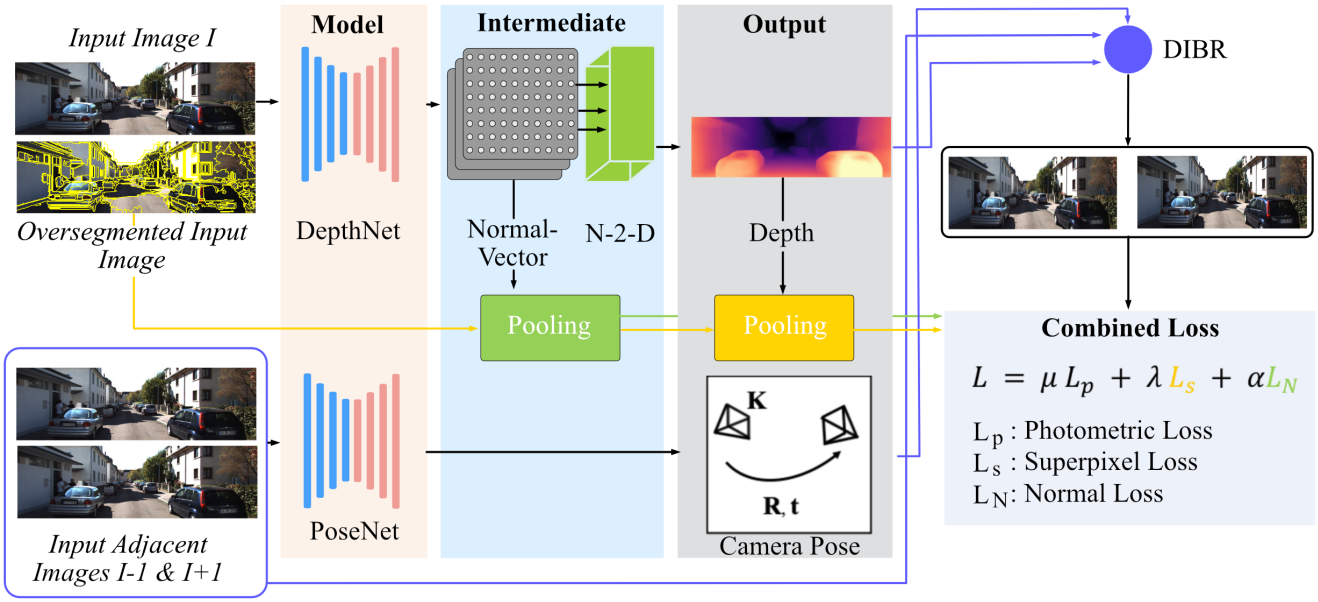

Figure 2. **Overview of the proposed framework.** Our model takes 4 channels as an input, split up into an input image I and a superpixel segmentation representation matrix and a pair of consecutive image frames $I+1$ and $I-1$. The information flow then consists of three elements: 1) providing the necessary information needed for the novel-view-synthesis approach as described in [16] [58] illustrated in purple. 2) The Normal-2-Depth block that takes 3 channels as input provided by the depth net. This normal vector is then pooled by the superpixel planar segments and included in the Combined Loss as Normal Loss; illustrated in green) 3) The depth maps itself are also pooled via the superpixel segments and included in the Combined Loss as Normal Loss, illustrated in green. *legend*: black arrow - indicate information flow, purple arrow: information related the DIBR, green arrow - normal, yellow - superpixel.

restoration. They compared the l_1-norm, l_2-norm, SSIM, MS-SSIM, and other metrics to train an image restoration network and concluded that a mixed approach based on l_1 and MS-SSIM leads to the best result. For computational reasons, the simple SSIM is used here.

Since the Novel-View-Synthesis is in a sense image restoration and the combination of the photometric error and the smoothness loss is a widely used and a proven loss function [16] [7] [17], we decided to maintain the basic structure of an SSIM and L1 loss and just modify the smoothness loss to incorporate superpixel information or expand the loss.

Inspired by Kwak *et al.* [22] who used a superpixel pooling layer inside the network for segmentation we aim to exploit the spatial information of the superpixel to enforce sharper edges. Instead of using the basic smoothness loss which weighs the disparity with $e^{-\frac{\delta I_x}{\delta x}}$, the disparity is weighted with a superpixel mask μ.

Given superpixel, described in section 3.2, the superpixel binary mask for horizontal (x) and vertical (y) direction is constructed as follows:

$$\mu_{x/y} = \begin{cases} 1, & \text{inside one plane} \\ 0, & \text{on a superpixel edge.} \end{cases} \tag{7}$$

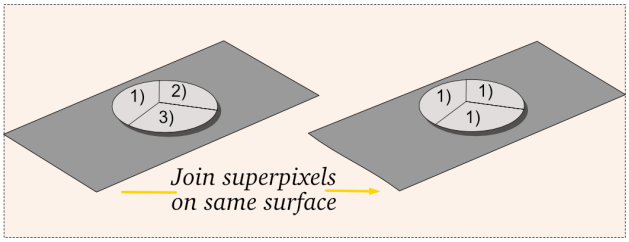

Figure 3. **Superpixel Loss.** The main disadvantage using oversegmentation is that one plane can be represented by multiple superpixel segments, so the superpixel needs to be joined by checking the disparity gradient in x and y direction in one plane. If high disparity gradient and superpixel segment changes occur, it can be assured it is a new surface and it is needed to enforce precise edges in the depth map at that particular pixel position.

In the superpixel continuous case the superpixel mask is constructed as:

$$\mu_{x/y} = \begin{cases} 1, & \text{inside Superpixel} \\ e^{-|\delta I_{x/y}|}, & \text{on an edge.} \end{cases} \tag{8}$$

Hence the new smoothness loss is defined as piecewise

$$L_s = |\delta_x d_t|\, \mu_x + |\delta_y d_t|\, \mu_y. \tag{9}$$

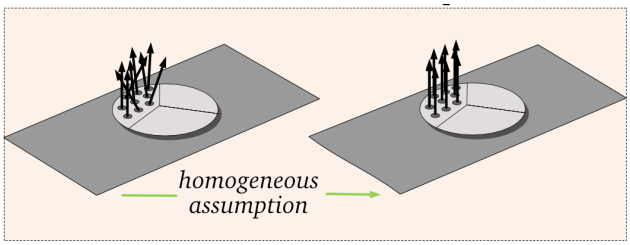

Figure 4. **Normal Loss.** It is assumed that a surface planar structure is best represented by normal on that plane that are as similar as possible.

In the binary case, the 1 inside one plane ensures that the gradient is minimized inside the plane. The 0 on the edges preserves that there is no minimizing therefore sharp edges. An intuition for this approach is illustrated in figure 3.

In the continuous case, the $e^{-|\delta I_{x/y}|}$ preserves sharpness on edges. So an implicit pooling is done, since the Superpixel Loss forces the disparity to be equal inside one superpixel, see the yellow box in figure 2.

3.4. Normal Loss

Our proposed framework calculates three plane coefficients (a_i, b_i, c_i) for each pixel i. However, the advantage of our approach is that the computed superpixels represent one plane each. Since all pixels in one superpixel area can be approximated as a plane, the three computed plane coefficients should be as similar as possible. This is homogeneous assumption is illustrated in figure 4. Therefore, the respective plane coefficients are averaged for each superpixel area s $(\bar{a_s}, \bar{b_s}, \bar{c_s})$, which is an implicit pooling operation, see figure 2. The sum of the deviation of the coefficient from the average of all pixels and superpixels then represents the Normal Loss L_n:

$$L_n = \alpha * \sum_{\mathrm{s}} \sum_{\mathrm{i} \in \mathrm{s}} |a_i - \bar{b_s}| + |b_i - \bar{b_s}| + |c_i - \bar{c_s}|, \quad (10)$$

where α is the weighting factor. The proposed Normal Loss is added to the loss function of the baseline, see section 3.5. One could think that the normals do not converge to a single value, but convergence is implicitly ensured by the other loss contributions as the normals are directly connected to disparity and depth, see section 3.1.

3.5. Combined Loss

As stated in sections 3.3 and 3.4 the Smoothness Loss of Monodepth2 is replaced by our proposed Superpixel Loss and our proposed Normal Loss is added to the overall loss function:

$$L = \mu L_p + \lambda L_s + \alpha L_n, \quad (11)$$

where L_s can be the Superpixel Loss or the baseline's Smoothness Loss, L_n is the Normal Loss, λ and α are scaling factors and μ is the automask presented by Monodepth2, see supplementary material 7.2.

4. Additional Considerations

Our network is based on Godard *et al*. [16]. We modified the depth net according to our proposed architecture, see figure 2. For training, we start with on ImageNet [36] pre-trained weights. Our model is trained for 20 epochs using Adam, a batch size of 12 and an image resolution of 640x192 pixel. The learning rate of 10^{-4} is reduced to 10^{-5} after 15 epochs. The smoothness term of the baseline's loss is fixed to $\lambda = 0.001$, while the weighting factor of the Normal Loss is determined to be $\alpha_n = 0.01$ (unless otherwise specified). All our models are trained with monocular images. For Felzenwalb's method, we use $scale = 120$, $\sigma = 0.8$ and a minimum size of 80 pixel. For SLIC, we use 100 segments, compactness of 0.9 and $\sigma = 80$.

5. Experiments

Here, we validate that (1) we modified the baseline successfully so that the network predicts geometric constraints instead of depth, (2) our proposed architecture performs well on the KITTI dataset. Third, we analyze the influence of the different possible variants of our architecture.

5.1. Surface Normal Coefficients

As proposed we aimed to learn plane coefficients representing unnormalized surface normals, see section 3.1. The network estimates the three coefficients a, b, c for every pixel. For further analyses, mean and standard deviation of the coefficients are calculated. The network is configured with a 4 channel input, the Normal-2-Depth block and the continuous Superpixel Loss.

Coefficient	Mean $\pm$ Standard deviation
a	0.0029 ± 0.005
b	0.61 ± 0.13
c	0.39 ± 0.191

Table 1. **Analysis of all normal coefficients for an example image.** The Mean and standard deviation of the output - normal coefficients, show that the corresponding normal vector n^T is not normalized to 1.

The a-values are around 0.0029, the b-values around 0.61 and the c-values around 0.39, see table 1. This shows that the vectors are not normalized to 1. Due to the lack of normalization, the learned normal coefficients don't correspond to real normal vectors. Additionally, the vectors of each superpixel seem to have nearly no extend in x-direction, because the a-value is very small.

To proof this assumption the superpixels were analyzed individually for mean, standard deviation, elevation, az-

imuth angle and magnitude. We decided to compare the normal coefficients for two different architectures. In the first configuration, the network has a 4 channel Input, the Normal-2-Depth block and the continuous Superpixel Loss (4ch + N2D + cont). In second configuration, the Normal Loss is added (4ch + N2D + cont + norm). The metrics for one superpixel are shown in table 2.

Parameter	Mean ± Std Config 1	Mean ± Std Config 2
a	0.007 ± 0.0039	-0.0009 ± 0.0098
b	0.695 ± 0.0279	0.67 ± 0.024
c	0.51 ± 0.016	0.47 ± 0.028
Magnitude r	0.74 ± 0.022	0.68 ± 0.026
Elevation θ	$46.62° \pm 1.8°$	$45.80° \pm 3.55°$
Azimuth φ	$0.58° \pm 0.32°$	$-0.08° \pm 0.84°$

Table 2. **Analysis of all normal vectors for one superpixel plane.** Mean, standard deviation, azimuth, and elevation angle for one arbitrary chosen superpixel are calculated. In the first configuration, the network has a 4 channel input, the Normal-2-Depth block and the continuous Superpixel Loss (Config 1). In the second configuration, the Normal Loss is added (Config 2).

In both cases there is indeed nearly no extension in x-direction, means the azimuth angle ($\approx 0,0007/ \approx -0.0009$) is close to $0°$, see table 2. The standard deviation of the elevation angle slightly increases, which suggests no improvement of the Normal Loss in terms of uniformly vectors inside one superpixel. Anyways every superpixel can nearly be represented by one vector, which is the case as the standard deviation of the shown metrics is low.

For a deeper understanding how the output looks like, the center of mass of each superpixel and the mean in x, y, z direction were calculated and one vector for each superpixel was depicted on top of an image during testing, see figure 5.

A comparison between disparity and the magnitude of the produced output shows that the depth/disparity information seems to be stored in the magnitude of the learned vectors, see figure 6.

5.2. KITTI Eigen-Zhou Split

We present results for the KITTI Eigen-Zhou split. KITTI dataset [13] contains 42,382 stereo pairs. The KITTI Eigen split proposed by Eigen *et al*. [9] contains 22600 training, 888 validation, and 697 test stereo image pairs. Zhou *et al*. [58] modified the Eigen split so it uses 39.600 training, 4424 validation and 1000 test images. All images are rectified and resized to 640 x 192 pixel. For training we use the KITTI Eigen-Zhou split and for testing the KITTI Eigen test set. For this comparison, we use a 4 channel input with superpixel calculated by Felzenwalb's method [31], the proposed Normal-2-Depth block, the continuous Superpixel Loss, and the Normal Loss. Comparing the depth map of our chosen configuration with the depth map of the approach from Godard *et al*. [16], one can see small differences, see figure 7. Our configuration represents surfaces more closed, while the approach by Godard *et al*. does not recognize the windows of the car as enclosed surfaces.

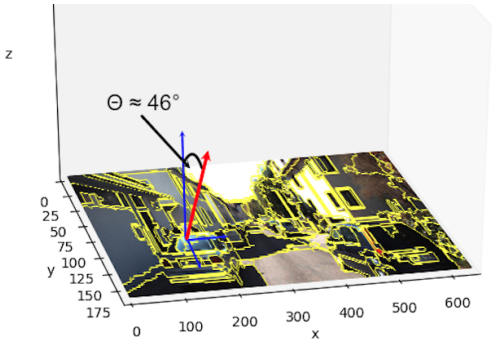

Figure 5. **Normal vector with 3D orientation.** We demonstrate an example normal vector with its according elevation angle on a segmented surface. The other properties for two different configurations are shown in table 2. For demonstration purposes, the vector is scaled arbitrarily.

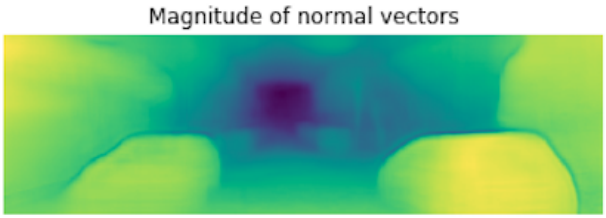

Figure 6. **Output Demonstration.** We show the magnitude of normal vectors for our network with a 4 channel input, the Normal-2-Depth block and the continuous Superpixel Loss (Config 1).

We evaluate the depth maps estimated using the metrics described in Eigen *et al*. [9] and compare them to other state-of-the-art approaches for monocular trained depth estimation, see table 3. We can estimate the depth reasonably well while we improve the representation of non-lombardian surfaces.

5.3. KITTI Ablation Study

To understand which components of our model are critical for depth estimation performance, we analyzed results on the KITTI 2015 dataset for a variety of possible configurations. We compare depth maps of three of our configurations with the depth map of Monodepth2 for three images, that represent typical scenes of KITTI, see figure 8. All our configurations tend to calculate non-lambertian surfaces like windows more homogeneous (car windows in the left and right image). The two configurations with superpixel information recognize a tree in the left image, which is not recognized in Monodepth2 and our configuration without

Method	Abs Rel	Sq Rel	RSME	RSME log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Zhou [58]	0.183	1.595	6.709	0.27	0.734	0.902	0.959
Yang [52]	0.182	1.481	6.501	0.267	0.725	0.906	0.963
Mahjourian [24]	0.163	1.24	6.22	0.25	0.762	0.916	0.968
GeoNet [56]	0.149	1.06	5.567	0.226	0.796	0.935	0.975
DDVO [45]	0.151	1.257	5.583	0.228	0.81	0.936	0.974
DF-Net [59]	0.15	1.124	5.507	0.223	0.806	0.933	0.973
LEGO [50]	0.162	1.352	6.276	0.252	-	-	-
Ranjan [33]	0.148	1.149	5.464	0.226	0.815	0.935	0.973
EPC++ [23]	0.141	1.029	5.35	0.216	0.816	0.941	0.976
Struct2depth [4]	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Monodepth2 [16]	0.115	0.903	4.863	0.193	0.877	0.959	0.981
Ours	0.141	1.276	5.549	0.221	0.832	0.941	0.974

Table 3. **Results on KITTI 2015 Eigen split [9].** All results are presented without post-processing. Results are taken from Godard *et al*. [16]. Our method here used a 4 channel input with superpixel calculated by Felzenwalb's method [31], the proposed Normal-2-Depth block, the continuous Superpixel Loss, and the Normal Loss. In the red columns, smaller values are better, in the blue columns larger values are better.

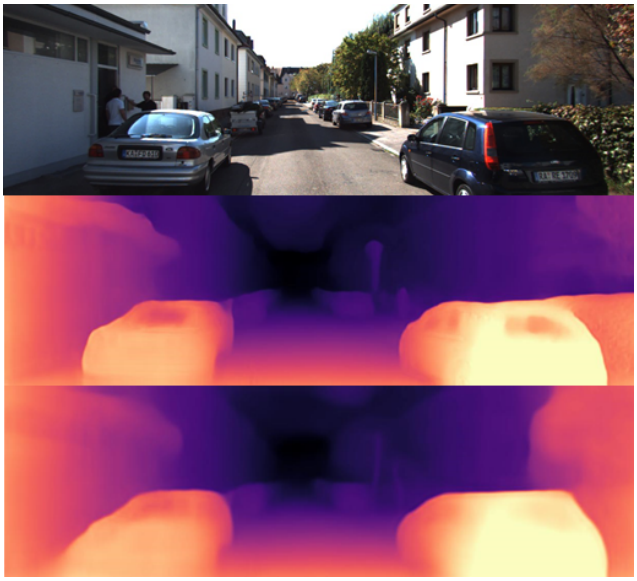

Figure 7. **Qualitative results on KITTI dataset.** From top to bottom: reference image and depth map output from Godard *et al*. [16] and ours. All images are shown without post-processing.

superpixel information. However, in the middle image, a thin tree stump in the left part of the image is no longer detected when using superpixel information. This could be due to the method of superpixel calculation. The influence of the method and also the weighting factor α of the Normal Loss is further analyzed, see table 4. It turns out SLIC does not achieve better results than Felzenswalb's method and and $\alpha = 0.01$ turns out to be a good weighting factor.

6. Conclusion

In this paper, we have proposed a modified framework of Monodepth2 [16] to improve monocular depth estimation and estimate surface normal coefficients of planar over segmented images via Felzenswalb's method and SLIC. The main contributions are the Normal-2-Depth block (N2D) and loss functions for including superpixel information and homogeneous normal approximation. An intense ablation study showed that not all combinations are showing promising results. A "4 channel input + N2D + continuous superpixel + normal-loss approach" showed the best result in terms of absolute relative error and visual examination of the created depth maps. It has to be clearly stated that the baseline architecture in terms of absolute relative error could not be improved through special case improvement in the created depth maps has been made. Further a feasibility study that normal coefficient can be estimated in a novel-view-synthesis workflow can be created, has been successfully established. The authors hope to inspire further research based on modeling a complex 3D scene via over-segmentation using superpixel as a piece-wise planar and rigid approximation and calculate surface normals for each planar structure inside a neural network architecture.

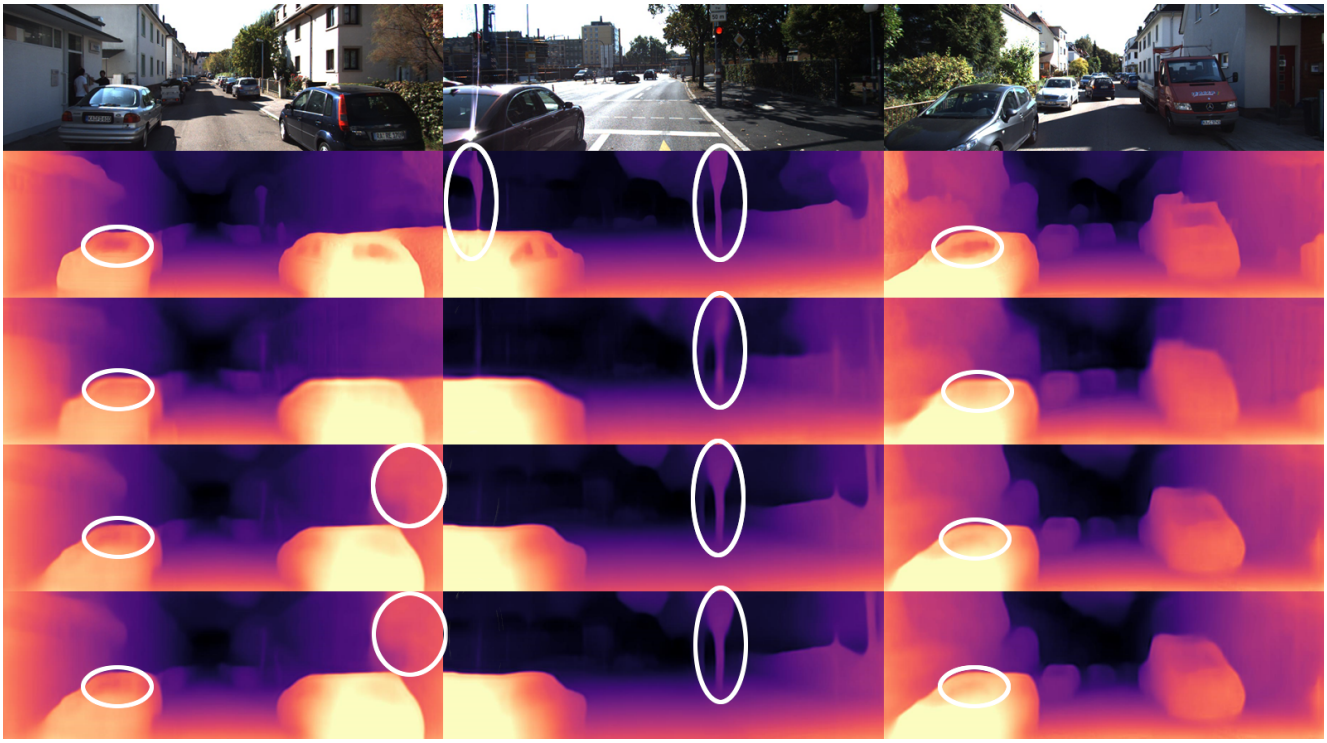

Figure 8. **Qualitative results for the ablation study.** From top to bottom: reference image, depth map output for the approach by Godard *et al*. [16], 4-Ch + N2D + cont + norm, 4-Ch + N2D + cont and 4-Ch + N2D. The methods using our proposed Normal-2-Depth block are denoted by N2D. The number of input channels (Ch) has been modified to 3, 4 or 6, see section 3.2. The superpixel are calculated with Felzenwalb's method [31].

Method	Decoder	Ch	Sup	Loss	Abs Rel	Sq Rel	RSME	RSMElog	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Monodepth2 [16]	standard	3		standard	0.115	0.903	4.863	0.193	0.877	0.959	0.981
N2D	normals	3		standard	0.123	0.984	5.042	0.2	0.859	0.955	0.98
4Ch	standard	4	fz	standard	0.141	1.313	5.545	0.22	0.834	0.942	0.974
4Ch	standard	4	sl	standard	0.255	2.237	7.892	0.342	0.594	9.832	0.927
6Ch	standard	6	fz	standard	0.122	0.978	5.026	0.2	0.862	0.955	0.979
4Ch + N2D	normals	4	fz	standard	0.142	1.262	5.551	0.22	0.83	0.942	0.975
4Ch + N2D + bin	normals	4	fz	binary	0.443	4.757	12.083	0.588	0.303	0.561	0.766
3Ch + N2D + bin	normals	3		binary	0.443	4.757	12.083	0.588	0.303	0.561	0.766
4Ch + N2D + cont	normals	4	fz	continuous	0.138	1.185	5.484	0.218	0.832	0.944	0.975
4Ch + N2D + cont	normals	4	sl	continuous	0.443	4.757	12.083	0.588	0.303	0.561	0.766
3Ch + N2D + cont	normals	3		continuous	0.443	4.757	12.083	0.588	0.303	0.561	0.766
4Ch + N2D + 0.001 norm	normals	4	fz	0.001 * norm	0.139	1.193	5.525	0.22	0.831	0.941	0.974
4Ch + N2D + 0.01 norm	normals	4	fz	0.01 * norm	0.139	1.172	5.513	0.218	0.831	0.942	0.975
4Ch + N2D + 0.1 norm	normals	4	fz	0.1 * norm	0.443	4.757	12.083	0.588	0.303	0.561	0.766
4Ch + N2D + bin + norm	normals	4	fz	bin + norm	0.443	4.757	12.083	0.588	0.303	0.561	0.766
4Ch + N2D + cont + norm	normals	4	fz	cont + norm	0.141	1.276	5.549	0.221	0.832	0.941	0.974
4Ch + N2D + cont + norm	normals	4	sl	cont + norm	0.443	4.757	12.083	0.588	0.303	0.561	0.766
3Ch + N2D + cont + norm	normals	3		cont + norm	0.443	4.757	12.083	0.588	0.303	0.561	0.766

Table 4. **Ablation.** Results for different variants of our model with monocular training on the KTTI Eigen [9] test set. All variants trained with the KITTI Eigen-Zhou split [58]. The methods using our proposed Normal-2-Depth block are denoted by N2D. The number of input channels (Ch) has been modified to 3, 4 or 6, see section 3.2. The method to calculate the superpixel used by the variant is denoted by fz for Felzenwalb's method [31] or by sl for SLIC [1]. In the red columns, smaller values are better, in the blue columns larger values are better.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels, 2010.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels

compared to state-of-the-art superpixel methods, 2012.

[3] Ali Borji, Ming-Ming Cheng, Huaizu Jiang, and Jia Li. Salient object detection: A survey. *Computational Visual Media*, 1411, 11 2014.

[4] Vincent Casser, Sören Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. *CoRR*, abs/1811.06152, 2018.

[5] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. *CoRR*, abs/1907.05820, 2019.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. *CoRR*, abs/1604.01685, 2016.

[7] Qi Dai, Vaishakh Patil, Simon Hecker, Dengxin Dai, and Konrad Schindler. Self-supervised object motion and depth estimation from video, 12 2019.

[8] Yan Di, Henrique Morimitsu, Shan Gao, and Xiangyang Ji. Monocular piecewise depth estimation in dynamic scenes by exploiting superpixel relations. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[9] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 2650–2658, USA, 2015. IEEE Computer Society.

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *CoRR*, abs/1406.2283, 2014.

[11] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *CoRR*, abs/1504.06852, 2015.

[12] Ravi Garg, Vijay Kumar B. G, and Ian D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *CoRR*, abs/1603.04992, 2016.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research*, 32(11):1231 – 1237, Sept. 2013.

[14] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

[16] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. October 2019.

[17] Rick Groenendijk, Sezer Karaoglu, T. Gevers, and Thomas Mensink. On the benefit of adversarial training for monocular depth estimation, 10 2019.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[19] K. Klasing, D. Althoff, D. Wollherr, and M. Buss. Comparison of surface normal estimation methods for range sensing applications. In *2009 IEEE International Conference on Robotics and Automation*, pages 3206–3211, 2009.

[20] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Monocular dense 3d reconstruction of a complex dynamic scene from two perspective frames. *CoRR*, abs/1708.04398, 2017.

[21] Suryansh Kumar, Ram Srivatsav Ghorakavi, Yuchao Dai, and Hongdong Li. A motion free approach to dense depth estimation in complex dynamic scene. *CoRR*, abs/1902.03791, 2019.

[22] Suha Kwak, Seunghoon Hong, and Bohyung Han. Weakly supervised semantic segmentation using superpixel pooling network, 2017.

[23] Chenxu Luo, Peng Wang, Yang Wang, Wei Xu, Ram Nevatia, and Alan Yuille. Every pixel counts ++: Joint learning of geometry and motion with 3d holistic understanding. 10 2018.

[24] Reza Mahjourian, Martin Wicke, and Anelia Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. *CoRR*, abs/1802.05522, 2018.

[25] Jyotsana Mehra and Nirvair Neeru. A brief review: Superpixel based image segmentation methods. 2016.

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis, 2020.

[27] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[28] P. Ndjiki-Nya, M. Koppel, D. Doshkov, H. Lakshman, P. Merkle, K. Muller, and T. Wiegand. Depth image-based rendering with advanced texture synthesis for 3-d video. *IEEE Transactions on Multimedia*, 13(3):453–465, June 2011.

[29] P. Neubert and P. Protzel. Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. In *2014 22nd International Conference on Pattern Recognition*, pages 996–1001, 2014.

[30] N.Mayer, E.Ilg, P.Häusser, P.Fischer, D.Cremers, A.Dosovitskiy, and T.Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. arXiv:1512.02134.

[31] Daniel P Huttenlocher Pedro F. Felzenswalb. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181., 2004.

[32] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 283–291, June 2018.

[33] Anurag Ranjan, Varun Jampani, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael J. Black. Adversarial collaboration: Joint unsupervised learning of depth, camera

motion, optical flow and motion segmentation. *CoRR*, abs/1805.09806, 2018.

[34] X. Ren and J. Malik. Learning a classification model for segmentation. volume Vol. 1, pages 10–17 vol.1, 11 2003.

[35] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014.

[37] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

[38] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, 2009.

[39] Mathijs Schuurmans, Maxim Berman, and Matthew B. Blaschko. Efficient semantic image segmentation with superpixel pooling. *CoRR*, abs/1806.02705, 2018.

[40] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. cite arxiv:1409.1556.

[41] Nouman Soomro and Murong Wang. Superpixel segmentation: A benchmark. *Signal Processing Image Communication*, 56, 04 2017.

[42] David Stutz. Superpixel segmentation: An evaluation. In Juergen Gall, Peter Gehler, and Bastian Leibe, editors, *Pattern Recognition*, volume 9358 of *Lecture Notes in Computer Science*, pages 555 – 562. Springer International Publishing, 2015.

[43] David Stutz, Alexander Hermans, and Bastian Leibe. Superpixels: An evaluation of the state-of-the-art. *Computer Vision and Image Understanding*, 166:1–27, 2018.

[44] T. Suzuki, S. Akizuki, N. Kato, and Y. Aoki. Superpixel convolution for segmentation. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 3249–3253, 2018.

[45] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. *CoRR*, abs/1712.00175, 2017.

[46] Murong Wang, Xiabi Liu, Yixuan Gao, Xiao Ma, and Nouman Soomro. Superpixel segmentation: A benchmark. *Signal Processing Image Communication*, 56:28–39., 04 2017.

[47] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *Trans. Img. Proc.*, 13(4):600–612, Apr. 2004.

[48] R. W. Wolcott and R. M. Eustice. Visual localization within lidar maps for automated urban driving. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 176–183, 2014.

[49] Junyuan Xie, Ross B. Girshick, and Ali Farhadi. Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks. *CoRR*, abs/1604.03650, 2016.

[50] Zhenheng Yang, Peng Wang, Yang Wang, Wei Xu, and Ram Nevatia. LEGO: learning edge with geometry all at once by watching videos. *CoRR*, abs/1803.05648, 2018.

[51] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *CoRR*, abs/1711.03665, 2017.

[52] Zhenheng Yang, Peng Wang, Wei Xu, Liang Zhao, and Ramakant Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *CoRR*, abs/1711.03665, 2017.

[53] Jian Yao, Marko Boben, Sanja Fidler, and Raquel Urtasun. Real-time coarse-to-fine topologically preserving segmentation. pages 2947–2955, 06 2015.

[54] Menglong Ye, Edward Johns, Ankur Handa, Lin Zhang, Philip Pratt, and Guang-Zhong Yang. Self-supervised siamese learning on stereo image pairs for depth estimation in robotic surgery, 2017.

[55] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. *CoRR*, abs/1907.12209, 2019.

[56] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. *CoRR*, abs/1803.02276, 2018.

[57] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on Computational Imaging*, 3(1):47–57, March 2017.

[58] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. *CoRR*, abs/1704.07813, 2017.

[59] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. *CoRR*, abs/1809.01649, 2018.

7. Supplementary Material

7.1. Visualization Monodepth2

Figure 9 is displayed on next page for better visibility.

7.2. Baseline's Loss

The baseline's loss is constructed by two parts. The first part is the minimum reprojection loss, which is inspired by Zhou *et al*. [58]. The reprojection loss computes the photometric error between two frames:

$$pe(I_a, I_b) = \frac{\alpha}{2}(1 - SSIM(I_a, I_b)) + (1-\alpha)\left\|I_a - I_b\right\|_1, \tag{12}$$

$\alpha = 0.85$. The baseline can be trained with stereo and monocular videos. Since we only deal with the monocular case, the image I_b is an inverse warped image from the two adjacent frames $I_{t'} \in \{I_{t-1}, I_{t+1}\}$ to the source frame I_t. Warping is done with information about depth D_t, Calibration Matrix K and the relative camera pose $T_{t\to t'}$ between the frames: $I_{t\to t'} = \langle proj(D_t, T_{t\to t'}, K)\rangle$, where $\langle\rangle$ is a sampling operator since the pixels enforce discrete values. The SSIM enhances the euclidean distance by taking human perception into account. The SSIM also incorporates luminance and contrast masking [47]. The human visual system is more sensitive to luminance and color variations in texture-less regions [57]. Since the Euclidean norm or l_2-norm is more sensitive to larger errors and more tolerant to small errors, equation 12 is based on a combination of the SSIM and the l_1-norm [57]. A significant improvement of Godard *et al*. [16] is taking the minimum reprojection Loss

$$L_p = min \quad pe(I_t, I_{t\to t'}). \tag{13}$$

The minimum photometric error prevents false depth predictions due to occlusions cause the photometric error is calculated between the previous and following frame and the smaller value is taken into account. It is likely that an occluded area is visible in one of the adjacent frames.

The second part of the loss is the smoothness loss, which has been presented in Godard *et al*. [15].

$$L_s = |\delta_x d_t|\, e^{-|\delta I_x|} + |\delta_y d_t|\, e^{-|\delta I_y|} \tag{14}$$

The smoothness loss ensures a smooth gradient of the disparity expect on edges. Thus the disparity is weighted with $e^{-|\delta I_{x/y}|}$.

The final loss is given by

$$L = \mu L_p + \lambda L_s. \tag{15}$$

where $\mu \in \{0, 1\}$ is a mask preventing stationary contaminating the loss [16] and $\lambda = 0.01$ a weighting factor. The loss is calculated every batch over multiple output scales upsampled to input resolution.

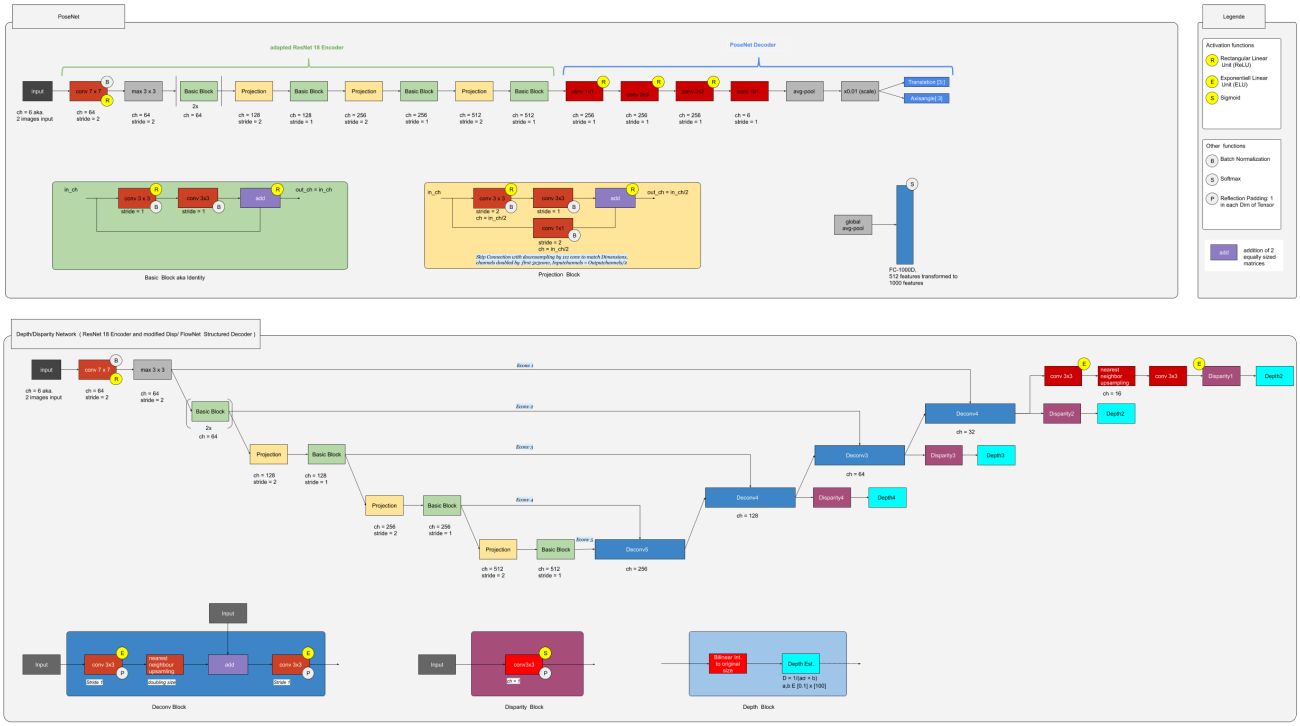

Figure 9. **Structure and Architecture of Monodepth2.** The figure shows the architecture of Monodepth 2 [16] visualized by the authors of this paper. The architecture of the depth network follows the DispNet-Structure [30] inspired by U-NET [35] and FlowNet [11]: an encoder decoder network with skip connections. The encoder of the network is by replaced by a simple Resnet18 CNN [18]. The decoder calculates the loss at four different output scales upsampled to original scale. Every convolution in the depth decoder is followed by an exponential linear unit (ELU). The encoder of the PoseNet is also a ResNet18 CNN modified to take a six channel Input (pretrained weights doubled for the additional channels). The pose decoder predicts the rotation (three euler angles) and the translation between the multiple input images (default = 2). For a detailed description, see Zhou *et al*. [58]. Typical neural networks building blocs are abbreviated as follows: yellow (R) = ReLU, (E) = ELU, (S) = sigmoid, grey: (B) = batch normalization, (S)= softmax, (P) = reflection padding and (ch) = output channels of the block (Please note the graphic in this paper is only for demonstration purpose. Please use the following link for high-resolution visualization: `https://tinyurl.com/vizMonodepth2`).