How Do Neural Networks See Depth in Single Images?

van Dijk, Tom; de Croon, Guido

DOI
10.1109/ICCV.2019.00227

Publication date
2019

Document Version
Final published version

Published in
Proceedings - 2019 International Conference on Computer Vision, ICCV 2019

How Do Neural Networks See Depth in Single Images?

Tom van Dijk
Technische Universiteit Delft
Delft, The Netherlands
J.C.vanDijk-1@tudelft.nl

Guido de Croon
Technische Universiteit Delft
Delft, The Netherlands
G.C.H.E.deCroon@tudelft.nl

Abstract

Deep neural networks have lead to a breakthrough in depth estimation from single images. Recent work shows that the quality of these estimations is rapidly increasing. It is clear that neural networks can see depth in single images. However, to the best of our knowledge, no work currently exists that analyzes what these networks have learned.

In this work we take four previously published networks and investigate what depth cues they exploit. We find that all networks ignore the apparent size of known obstacles in favor of their vertical position in the image. The use of the vertical position requires the camera pose to be known; however, we find that these networks only partially recognize changes in camera pitch and roll angles. Small changes in camera pitch are shown to disturb the estimated distance towards obstacles. The use of the vertical image position allows the networks to estimate depth towards arbitrary obstacles – even those not appearing in the training set – but may depend on features that are not universally present.

1. Introduction

Stereo vision allows absolute depth to be estimated using multiple cameras. When only a single camera can be used, optical flow can provide a measure of depth; or if images can be combined over longer time spans then SLAM or Structure-from-Motion can be used to estimate the geometry of the scene. These methods tend to treat depth estimation as a purely geometrical problem, ignoring the *content* of the images.

When only a *single* image is available, it is not possible to use epipolar geometry. Instead, algorithms have to rely on *pictorial cues*: cues that indicate depth within a single image, such as texture gradients or the apparent size of known objects. Shape-from-X methods (e.g. [18, 1], [14], [7]) use some of these cues to infer shape, but often make strong assumptions that make them difficult to use in unstructured environments such as those seen in autonomous driving. Other cues such as the apparent size of objects may require knowledge about the environment that is difficult to program by hand. As a result, pictorial cues have seen relatively little use in these scenarios until recently.

With the arrival of stronger hardware and better machine-learning techniques – most notably Convolutional Neural Networks (CNN) – it is now possible to *learn* pictorial cues rather than program them by hand. One of the earliest examples of monocular depth estimation using machine learning was published in 2006 by Saxena *et al*. [19]. In 2014, Eigen *et al*. [4] were the first to use a CNN for monocular depth estimation. Where [4] still required a true depth map for training, in 2016 Garg *et al*. proposed a new scheme that allows the network to learn directly from stereo pairs instead [9]; this work was further improved upon by Godard *et al*. in [11]. In parallel, methods have been developed that use monocular image sequences to learn single-frame depth estimation in an unsupervised manner, of which the works by Zhou *et al*. [25] and Wang *et al*. [23] are examples. Recent work focuses primarily on the accuracy of monocular depth estimation, where evaluations on publicly available datasets such as KITTI [15] and NYUv2 [20] show that neural networks *can* indeed generate accurate depth maps from single images. However, to the best of our knowledge, no work exists that investigates *how* they do this.

Why is it important to know what these neural networks have learned? Firstly, it is difficult to guarantee correct behavior without knowing what the network does. Evaluation on a test set shows that it works correctly in those cases, but it does not guarantee correct behavior in other scenarios. Secondly, knowing what the network has learned provides insight into training. Additional guidelines for the training set and data augmentation may be derived from the learned behavior. Thirdly, it provides insight into transfer to other setups. With an understanding of the network, it is for instance easier to predict what the impact of a change in camera height will be and whether this will work out-of-the-box, require data augmentation or even a new training set.

In this work, we take four previously published neural networks (MonoDepth by Godard *et al*. [11], SfMLearner

by Zhou *et al*. [25], Semodepth by Kuznietsov *et al*. [13] and LKVOLearner by Wang *et al*. [23]) and investigate their *high-level behavior*, where we focus on the distance estimation towards cars and other obstacles in an autonomous driving scenario. Section 2 gives an overview of related literature. In section 3 we show that the all of the networks rely on the vertical image position of obstacles but not on their apparent size. Using the vertical position requires knowledge of the camera pose; in section 4 we investigate whether the camera pose is assumed constant or observed from the images. For MonoDepth we investigate in section 5 how it recognizes obstacles and finds their ground contact point. We discuss the impact of our results in section 6.

2. Related work

Existing work on monocular depth estimation has extensively shown that neural networks can estimate depth from single images, but an analysis of how this estimation works is still missing. Feature visualization and attribution could be used to analyze this behavior. One of the earlier examples of feature visualization in deep networks can be found in [6]. The methods have been improved upon in e.g. [22, 24] and an extensive treatment of visualization techniques can be found in [16]. In essence, the features used by a neural network can be visualized by optimizing the input images with respect to a loss function based on the excitation of a single neuron, a feature map or an entire layer of the network. The concurrent work of Hu *et al*. [12] in which the authors perform an attribution analysis to find the pixels that contribute most to the resulting depth map is most closely related to our work. However, these methods only provide insight into the low-level workings of CNNs. A collection of features that the neural network is sensitive to is not a full explanation of its behavior. A link back to depth cues and behavior in more human terms is still missing, which makes it difficult to reason about these networks.

In this work, we take a different approach that is perhaps more closely related to the study of (monocular) depth perception in humans. We treat the neural network as a black box, only measuring the response (in this case depth maps) to certain inputs. Rather than optimizing the inputs with regards to a loss function, we modify or disturb the images and look for a correlation in the resulting depth maps.

Literature on human depth perception provides insight into the pictorial cues that could be used to estimate distance. The following cues from [10] and more recent reviews [3, 2] can typically be found in single images:

- Position in the image. Objects that are further away tend to be closer to the horizon. When resting on the ground, the objects also appear higher in the image.
- Occlusion. Objects that are closer occlude those that lie behind them. Occlusion provides information on depth order, but not distance.
- Texture density. Textured surfaces that are further away appear more fine-grained in the image.
- Linear perspective. Straight, parallel lines in the physical world appear to converge in the image.
- Apparent size of objects. Objects that are further away appear smaller.
- Shading and illumination. Surfaces appear brighter when their normal points towards a light source. Light is often assumed to come from above. Shading typically provides information on depth changes within a surface, rather than relative to other parts of the image.
- Focus blur. Objects that lie in front or behind the focal plane appear blurred.
- Aerial perspective. Very far away objects (kilometers) have less contrast and take on a blueish tint.

Of these cues, we expect that only the *position in the image* and *apparent size of objects* are applicable to the KITTI dataset; other cues are unlikely to appear because of low image resolution (texture density, focus blur), limited depth range (aerial perspective) or they are less relevant for distance estimation towards obstacles (occlusion, linear perspective and shading and illumination).

Both cues have been experimentally observed in humans, also under conflicting conditions. Especially the vertical position in the visual field has some important nuances. For instance, Epstein shows that perceived distances do not solely depend on the vertical position in the visual field, but also on the background [5]. Another important contextual feature is the horizon line, which gains further importance when little ground (or ceiling) texture is present [8]. Using prismatic glasses that manipulated the human subjects' vision, Ooi *et al*. showed that humans in real-world experiments use the angular declination relative to the 'eye level' [17] rather than the visual horizon, where the eye level is the *expected* height of the horizon in the visual field. The apparent size of objects also influences their estimated distance. Sousa *et al*. performed an experiment where subjects needed to judge distances to differently-sized cubes [21]. The apparent size of the cubes influenced the estimated distance even though the true size of the cubes was not known and the height in the visual field and other cues were present. No work was found that investigates whether these observations also apply to neural networks for depth estimation.

3. Position vs. apparent size

As stated in section 2, the vertical image position and apparent size of objects are the most likely cues to be used by the networks. Figure 1 shows how these cues can be used to estimate the distance towards obstacles. The camera's focal length f is assumed known and constant and is implicitly learned by the neural network. We furthermore assume that the camera's pitch angle relative to the horizon is

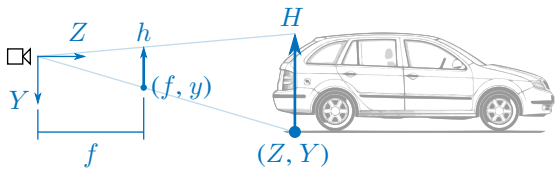

Figure 1. True object size H and position Y, Z in the camera frame and vertical image position y and apparent size h in image coordinates. Image coordinates are measured from the center of the image.

small; pitch angles can therefore be approximated by a shift in vertical image coordinates y, where the horizon level y_h is used as a measure for the camera's pitch. All coordinates are measured relative to the center of the image.

Given the obstacle's real-world size H and apparent size h in the image, the distance can be estimated using:

$$Z = \frac{f}{h} H \tag{1}$$

This requires the obstacle's true size H to be known. The objects encountered most often in the KITTI dataset come from a limited number of classes (e.g. cars, trucks, pedestrians), where all objects within a class have roughly the same size. It is therefore possible that the networks have learned to recognize these objects and use their apparent size to estimate their distance.

Alternatively, the networks could use the vertical image position y of the object's ground contact point to estimate depth. Given the height Y of the camera above the ground, the distance can be estimated through:

$$Z = \frac{f}{y - y_h} Y \tag{2}$$

This method does not require any knowledge about the true size H of the object, but instead assumes the presence of a flat ground and known camera pose $(Y,\, y_h)$. These assumptions also approximately hold in the KITTI dataset.

3.1. Evaluation method

To find which of these cues are used by the networks, three sets of test images are generated: one in which the apparent size of objects is varied but the vertical position of the ground contact point in the image is kept constant, one in which the vertical position is varied but the size remains constant, and a control set in which both the apparent size and position are varied with distance – as would be expected in real-world images.

The test images are generated as follows: the objects (mostly cars) are cropped from the images of KITTI's scene flow dataset. Each object is labeled with its location relative to the camera (e.g. one lane to the left, facing towards the camera) and with the position in the image it was cropped from. Secondly, each image in the test set was labeled with positions where an obstacle could be inserted (e.g. the lane to the left of the camera is still empty). Combining this information with the object labels ensures that the test images remain plausible.

The true distance to the inserted objects is not known; instead the network's ability to measure relative distances will be evaluated. Distances are expressed in relation to the original size and position of the object, which is assigned a relative distance $Z'/Z = 1.0$. The relative distance is increased in steps of 0.1 up to 3.0 and controls the scaling s and position x', y' of the object as follows:

$$s = \frac{Z}{Z'}, \tag{3}$$

and

$$x' = x\frac{Z}{Z'}, \quad y' = y_h + (y - y_h)\frac{Z}{Z'} \tag{4}$$

with x', y' the new coordinates of the ground contact point of the object and with y_h the height of the horizon in the image which is assumed constant throughout the dataset.

The estimated depth towards the car is evaluated by averaging the depth map over a flat region on the front or rear of the car (Figure 2). A flat region is used rather than the entire object to prevent the estimated length of the vehicle from influencing the depth estimate; the length is very likely dependent on the apparent size of the object, while the distance might not be.

3.2. Results

The results of this experiment are shown in Figure 3. When both the position and scale are varied, all depth estimates except Wang *et al.*'s behave as expected: the estimated depth stays close to the true depth of the object which shows that the networks still work correctly on these artificial images. When only the vertical position is varied, the networks can still roughly estimate the distance towards the objects, although this distance is slightly overestimated (Godard *et al.*, Zhou *et al.*, Wang *et al.*) or underestimated (Kuznietsov *et al.*). Additionally, the standard deviation of the distance estimate has increased compared to the control set. The most surprising result is found when only the apparent size of the object is changed but the ground contact point is kept constant: none of the networks observe any change in distance under these circumstances.

These results suggest that the neural networks rely primarily on the vertical position of objects rather than their apparent size, although some change in behavior is observed when the size information is removed. The fact that all four networks show similar behavior also suggests that this is a general property that does not strongly depend on the network architecture or training regime (semi-supervised, unsupervised from stereo, unsupervised from video).

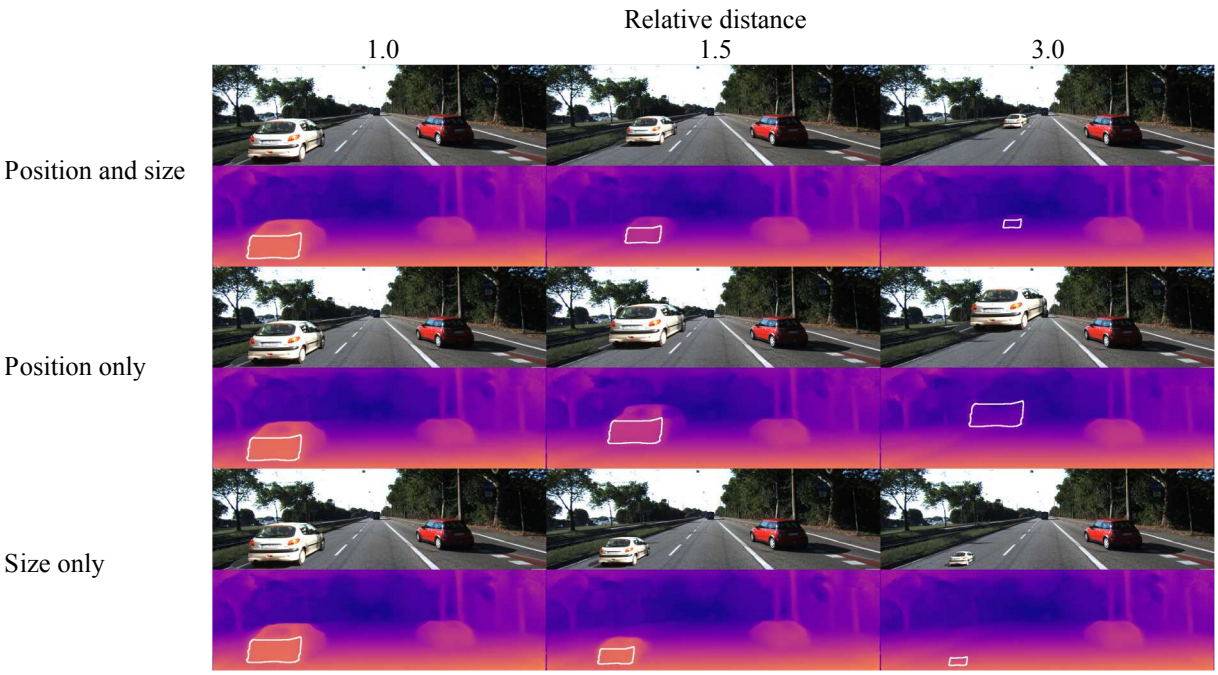

Figure 2. Example test images and resulting disparity maps from MonoDepth. The white car on the left is inserted into the image at a relative distance of 1.0 (left column), 1.5 (middle column) and 3.0 (right column), where a distance of 1.0 corresponds to the size and position at which the car was cropped from its original image. In the top row, both the position and apparent size of the car vary with distance, in the middle row only the position changes and the size is kept constant, and in the bottom row the size is varied while the position is constant. The region where the estimated distance is measured is indicated by a white outline in the disparity maps.

4. Camera pose: constant or estimated?

The use of vertical position as a depth cue implies that the networks have some knowledge of the camera's pose. This pose could be inferred from the images (for instance, by finding the horizon or vanishing points), or assumed to be constant. The latter assumption should work reasonably well on the KITTI dataset, where the camera is rigidly fixed to the car and the only deviations come from pitch and heave motions of the car and from slopes in the terrain. It would, however, also mean that the trained networks cannot be directly transferred to different camera setups. It is therefore important to investigate whether the networks assume a fixed camera pose or estimate this on-the-fly.

If the networks can measure the camera pitch, then changes in pitch should also be observed in the estimated depth map. The unmodified KITTI test images already have some variation in the horizon level; in an initial experiment we look for a correlation between the true horizon level in the images (determined from the Velodyne data) and the estimated horizon level in the depth estimates from MonoDepth. The horizon levels were measured by cropping a central region of the disparity map (the road surface) and using RANSAC to fit a line to the disparity-y pairs. Extrapolating this line to a disparity of zero (i.e. infinite distance) gives the elevation of the horizon. For each image, this procedure was repeated five times to average out fitting errors from the RANSAC procedure.

Figure 4 shows the relation between the true and estimated horizon levels. While it was expected that MonoDepth would either fully track the horizon level or not at all, a regression coefficient of 0.60 was found which indicates that it does something between these extremes.

A second experiment was performed to rule out any potential issues with the Velodyne data and with the small (± 10 px) range of true horizon levels in the first experiment. In this second experiment, a smaller region is cropped at different heights in the image (Figure 5). For each image, seven crops are made with offsets between -30 and 30 pixels from the image center, which approximates a change in camera pitch of $\pm$2-3 degrees. Instead of using the Velodyne data to estimate the true horizon level, the horizon level from the depth estimate of the centrally cropped image is used as a reference value. In other words, this experiment evaluates how well a *shift* in the horizon level is reflected in the depth estimate, rather than its absolute position.

The results for all four networks are shown in Figure 6. A similar result as in the previous experiment is found: all networks are able to detect changes in camera pitch, but the change in the horizon level is underestimated by all networks. Since the networks use the vertical position of obstacles to estimate depth, we expect this underestimation to af-

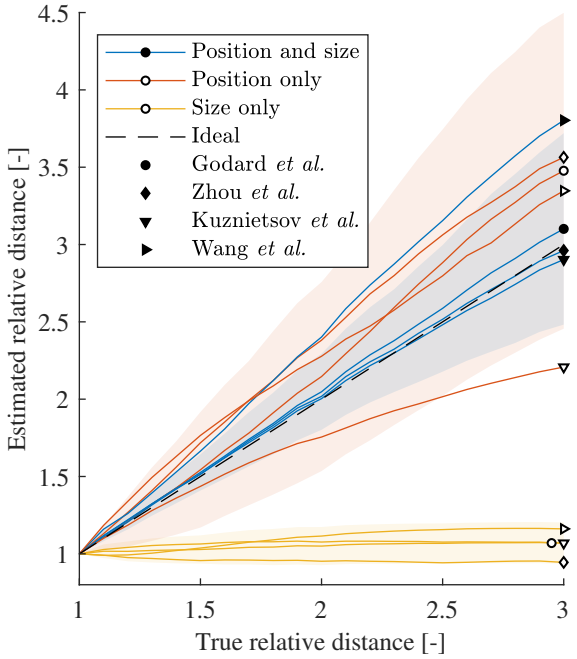

Figure 3. Influence of vertical image position and apparent size cues on depth estimates. Shaded regions indicate ±1 SD (N = 1862) for the network by Godard *et al*. When both depth cues are present, all networks successfully estimate the distance towards the objects, except Wang *et al*.'s which overestimates the distance. When only the vertical position is available, the distance is either over- or underestimated and the standard deviation of the measurement increases (only shown for MonoDepth). When only the apparent size is available, none of the networks are able to estimate distance.

fect the estimated distances. To test this hypothesis, we use the same pitch crop dataset and evaluate whether a change in camera pitch causes a change in obstacle disparities. The results are shown in Figure 7. The estimated disparities are indeed affected by camera pitch. This result also suggests that the networks look at the vertical image position of objects rather than their distance to the horizon, since the latter does not change when the images are cropped.

4.1. Camera roll

Similarly to the pitch angle, the roll angle of the camera influences the depth estimate towards obstacles. If the camera has a nonzero roll angle, the distance towards obstacles does not only depend on their vertical position but also on their horizontal position in the image. A similar experiment was performed as for the pitch angle: a smaller region of the images was cropped at varying angles (Figure 8). To measure the roll angle, a Hough line detector was applied to a thin slice of the depth map to find the angle of the road surface. As in the previous experiment, we look for a correlation between the camera angle and the *change* in the estimated angle of the road surface. The result is shown in Figure 9 and is similar to that for pitch angles: all networks

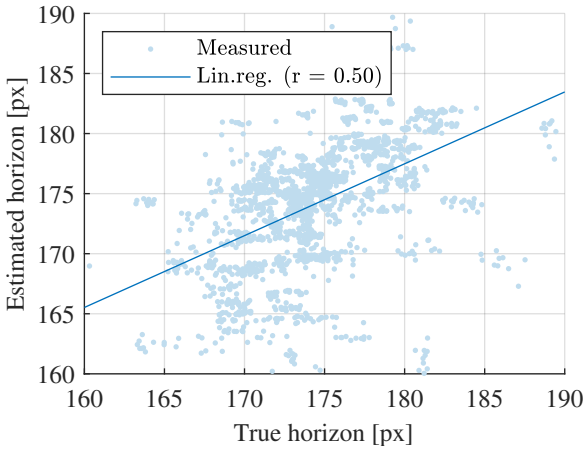

Figure 4. True and estimated horizon levels in unmodified KITTI images. Results for MonoDepth (Godard *et al*.). A medium-to-large correlation is found (Pearson's r = 0.50, N = 1892) but the slope is only 0.60, indicating that the true shift in the horizon is not fully reflected in the estimated depth map.

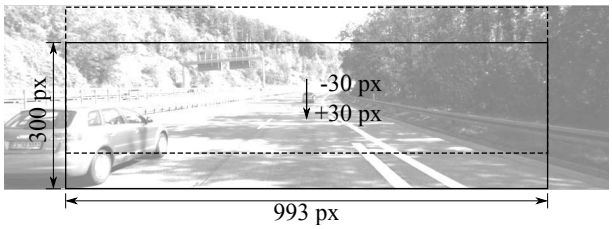

Figure 5. Larger camera pitch angles are emulated by cropping the image at different heights.

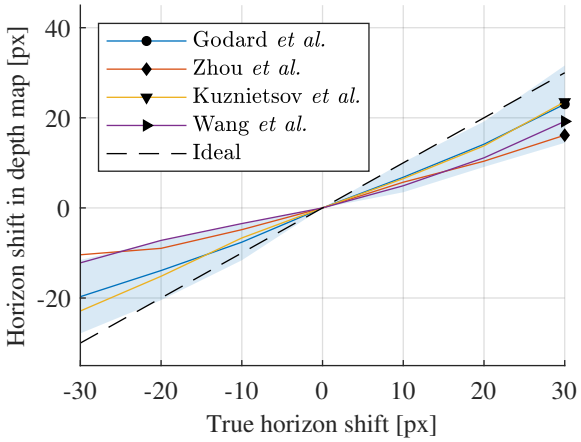

Figure 6. True and estimated shifts in horizon levels after cropping the images at different heights. Shaded regions indicate ±1 SD for the network by Godard *et al*. (N = 194, six outliers > 3 SD removed).

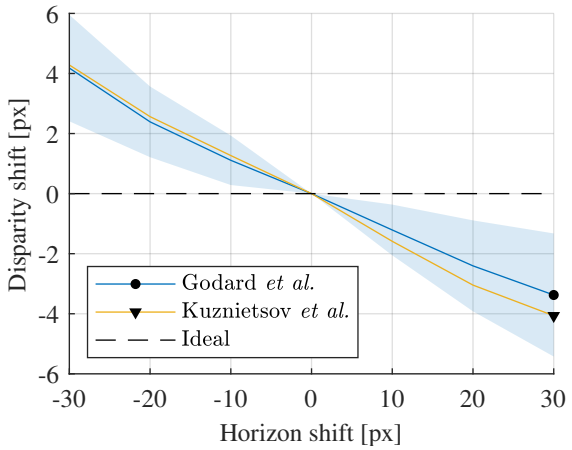

Figure 7. Changes in camera pitch disturb the estimated distance towards obstacles. Shaded regions indicate ± 1 SD for the network by Godard *et al*.

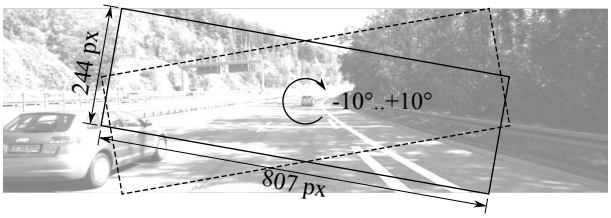

Figure 8. Camera roll angles are emulated by cropping smaller, tilted regions from the original KITTI images.

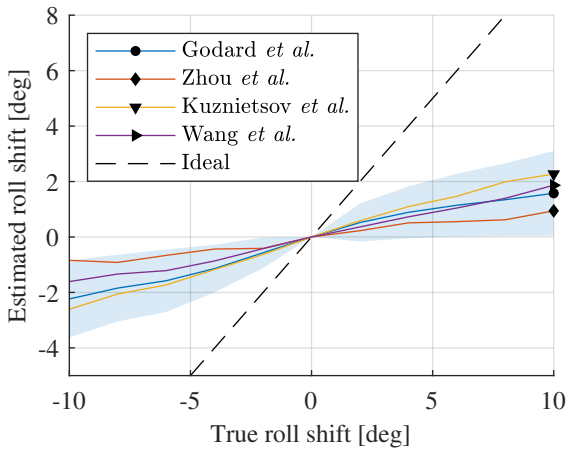

Figure 9. True and estimated roll shifts in the cropped images. For all networks, the change in road surface angle is smaller than the true angle at which the images are cropped. Shaded regions indicate ± 1 SD for the network by Godard *et al*. ($N = 189$, eleven outliers > 3 SD removed).

are able to detect a roll angle for the camera, but the angle is underestimated.

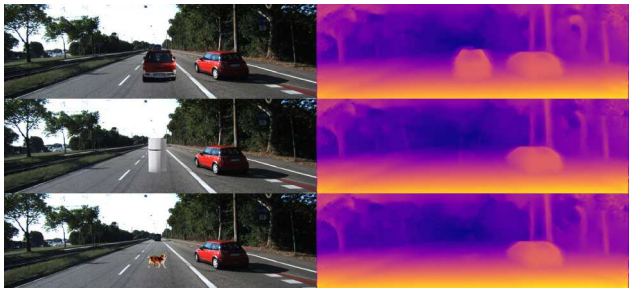

Figure 10. Objects that are not found in the training set (fridge, dog) are not reliably detected when pasted into the image.

5. Obstacle recognition

Section 3 has shown that all four networks use the vertical position of objects in the image to estimate their distance. The only knowledge that is required for this estimate is the location of the object's ground contact point. Since no other knowledge about the obstacle is required (e.g. its real-world size), this suggests that the networks can estimate the distance towards arbitrary obstacles. Figure 10, however, shows that this is not always the case. The car is recognized as an obstacle, but the other objects are not recognized and appear in the depth map as a flat road surface.

To correctly estimate the depth of an obstacle, the neural networks should be able to: 1) find the ground contact point of the obstacle, as this is used to estimate its distance, and 2) find the outline of the obstacle in order to fill the corresponding region in the depth map. In this section, we attempt to identify the features that the MonoDepth network by Godard *et al*. uses to perform these tasks. The results of Figure 10 suggest that the network relies on features that are applicable to cars but not to the other objects inserted into the test images.

5.1. Color and Texture

The objects inserted in Figure 10 differ from cars in terms of color, texture and shape. In a first experiment, we investigate how color and texture affect MonoDepth's performance by evaluating it on modified versions of the KITTI images. To investigate the influence of color, two new test sets are created: one in which the images are converted to grayscale to remove all color information, and one in which the hue and saturation channels are replaced by those from KITTI's semantic_rgb dataset to further disturb the colors. Two other datasets are used to test the influence of texture: a set in which all objects are replaced by a flat color that is the average of that object class – removing all texture but keeping the color intact – and the semantic_rgb set itself where objects are replaced with unrealistic flat colors. Examples of the modified images and resulting depth maps are shown in Figure 11, the performance measures are listed in Table 1.

As long as the value information in the images remains

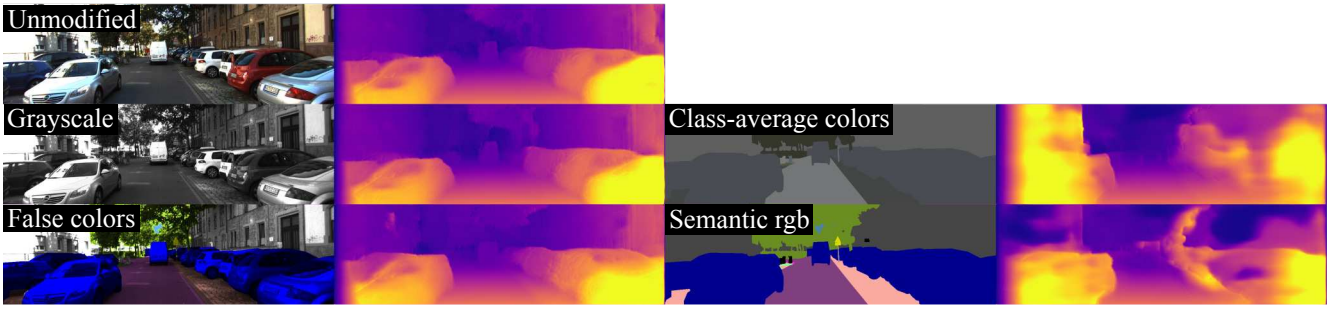

Figure 11. Example images and depth maps for unmodified, grayscale, false color, class average color, and semantic rgb images.

Test set	Abs Rel	Sq Rel	RMSE	RMSE log	D1-all	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Unmodified images	**0.124**	1.388	**6.125**	**0.217**	**30.272**	**0.841**	**0.936**	**0.975**
Grayscale	0.130	1.457	6.350	0.227	31.975	0.831	0.930	0.972
False colors	0.128	**1.257**	6.355	0.237	34.865	0.816	0.920	0.966
Semantic rgb	0.192	2.784	8.531	0.349	46.317	0.714	0.850	0.918
Class-average colors	0.244	4.159	9.392	0.367	50.003	0.691	0.835	0.910

Table 1. MonoDepth's performance on images with disturbed colors or texture. The unmodified image results were copied from [11]; the table lists results without post-processing. Error values for images that keep the value channel intact (*grayscale* and *false colors*) are close to the unmodified values. Images where the value information is removed and the objects are replaced with flat colors (*semantic rgb*, *class-average colors*) perform significantly worse.

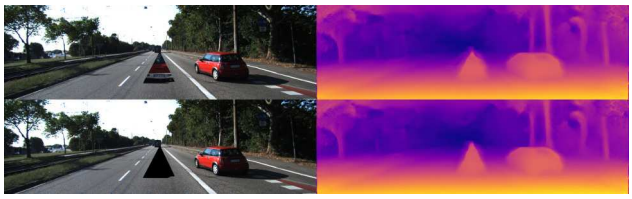

Figure 12. Objects do not need to have a familiar shape nor texture to be detected. The distance towards these non-existent obstacles appears to be determined by the position of their lower extent.

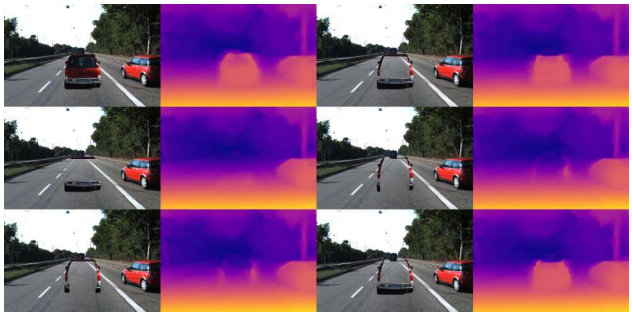

Figure 13. Influence of car parts and edges on the depth map. Removing the center of the car (top right) has no significant influence on the detection. The car's bottom and side edges (bottom right) seem most influential for the detected shape, which is almost identical to the full car image (top left).

unchanged (the *unmodified*, *grayscale* and *false color* images), only a slight degradation in performance is observed. This suggests that the exact color of obstacles does not strongly affect the depth estimate. However, when the texture is removed (*class-average colors* and *semantic rgb*) the performance drops considerably. The network also performs better on the semantic_rgb dataset with false colors than on the realistically colored images. This further suggests that the exact color of objects does not matter and that features such as the contrast between adjacent regions or bright and dark regions within objects are more important.

5.2. Shape and contrast

Since color does not explain why the objects in Figure 10 are not detected, we next look at shape and contrast. A first qualitative experiment shows that objects do not need a familiar shape nor texture to be recognized (Figure 12). Furthermore, the distance towards these unfamiliar objects seems to follow from their lower extent, further supporting our claim that the networks use the ground contact point as the primary depth cue.

In a second experiment, we find the features that the network is the most sensitive to by systematically removing parts of a car until it is no longer detected. The car is still detected when the interior of the shape is removed, suggesting that the network is primarily sensitive to the outline of the object and 'fills in' the rest. The car is no longer detected when the side- or bottom edges are removed. However, when only the bottom edge is removed, the sides of the car are still detected as two thin objects.

We suspect that the dark region at the bottom of the shape is the main feature by which the network detects obstacles.

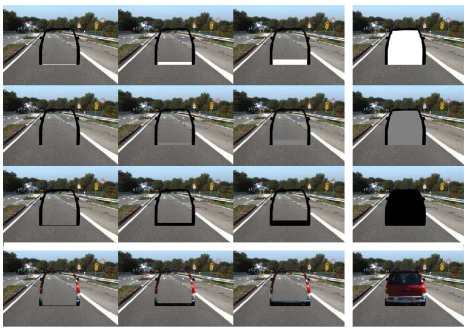

Figure 14. To measure the influence of the bottom edge, we vary its brightness and thickness. The experiment is repeated over 60 background images.

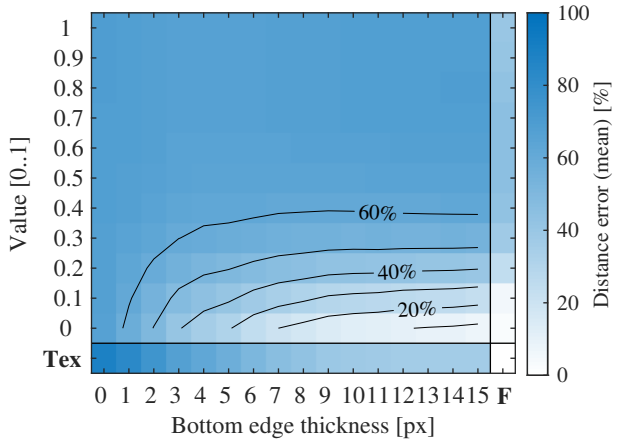

Figure 15. Mean distance error as a function of the bottom edge color and thickness. For comparison, we include results for realistically textured shapes (**Tex**) and completely filled shapes (**F**). Distance errors are measured relative to the estimated distance of the realistic (**F**, **Tex**) shape.

The bottom edge formed by the shadow underneath the car is highly contrasting with the rest of the scene and an examination of the KITTI images shows that this shadow is almost universally present and could form a reliable feature for the detection of cars. We examine the influence of the bottom edge in a quantitative experiment where both the brightness and thickness of the lower edge are varied (Figure 14, 15). Additionally, the results are compared to completely filled shapes (**F**) and shapes with a realistic texture (**Tex**). We measure the error in the obstacle's depth relative to the estimated depth of the realistic shape (**F**, **Tex**). The results are averaged over 60 background images in which the shape does not overlap other cars.

Figure 15 shows that the bottom edge needs to be both thick and dark for a successful detection, where a completely black edge with a thickness of ≥ 13 px leads to an average distance error of less than 10% relative to a realistic image. A white edge does not result in a successful detection despite having a similar contrast to the road surface.

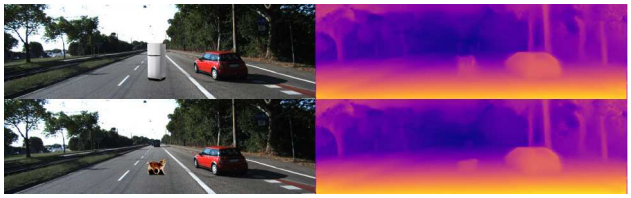

Figure 16. Adding a shadow at the bottom of the objects of Figure 10 causes them to be detected. The fridge, however, is only detected up to the next horizontal edge between the doors.

Furthermore, a completely black edge results in a smaller distance error than when realistic textures are used. This suggests that the network is primarily looking for a dark color rather than contrast or a recognizable texture. Finally, the results show that completely filled shapes result in a better distance estimate. We suspect that completely filling the shape removes edges from the environment that could otherwise be mistaken for the outline of the obstacle.

As a final test, we add a black shadow to the undetected objects of Figure 10. The objects are now successfully detected (Figure 16).

6. Conclusions and future work

In this work we have analyzed four neural networks for monocular depth estimation and found that all of them use the vertical position of objects in the image to estimate their depth, rather than their apparent size. This estimate depends on the pose of the camera, but changes to this pose are not fully accounted for, leading to an under- or overestimation of the distance towards obstacles when the camera pose changes. This limitation has a large impact on the deployment of these systems, but has so far received hardly any attention in literature. We further show that MonoDepth can detect objects that do not appear in the training set, but that this detection is not always reliable and depends on factors such as the presence of a shadow under the object.

While our work shows *how* these neural networks perceive depth, it does not show where this behavior comes from. Likely causes are the lack of variation in the training set, which could be corrected by data augmentation, or properties inherent to convolutional neural networks (e.g. their invariance to translation but not to scale). Future work should investigate which of these is true and whether the networks can learn to use different depth cues when the vertical image position is no longer reliable.

Acknowledgements We would like to thank the authors of [11, 25, 13, 23] for making their code and models publicly available. This work was performed as part of the Percevite project, funded by the SESAR Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 763702.

References

[1] Jonathan T. Barron and Jitendra Malik. Shape, Illumination, and Reflectance from Shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(8):1670–1687, aug 2015. 1

[2] Eli Brenner and Jeroen B.J. Smeets. Depth Perception. In J.T. Wixted, editor, *Stevens' Handbook of Experimental Psychology and Cognitive Neuroscience*, chapter Depth Perc, pages 385–414. John Wiley & Sons, New York, 4 edition, 2018. 2

[3] James E. Cutting and Peter M. Vishton. Perceiving Layout and Knowing Distances: The Integration, Relative Potency, and Contextual Use of Different Information about Depth. In *Perception of Space and Motion*, pages 69–117. Elsevier, 1995. 2

[4] David Eigen, Christian Puhrsch, and Rob Fergus. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. In *Advances in Neural Information Processing Systems 27*, pages 2366–2374. Curran Associates, Inc., 2014. 1

[5] William Epstein. Perceived Depth as a Function of Relative Height under Three Background Conditions. *Journal of Experimental Psychology*, 72(3):335–338, 1966. 2

[6] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network, 2009. 2

[7] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):406–417, mar 2005. 1

[8] Jonathan S. Gardner, Joseph L. Austerweil, and Stephen E. Palmer. Vertical position as a cue to pictorial depth: Height in the picture plane versus distance to the horizon. *Attention, Perception, & Psychophysics*, 72(2):445–453, 2010. 2

[9] Ravi Garg, Vijay B.G. Kumar, Gustavo Carneiro, and Ian Reid. Unsupervised CNN for Single View Depth Estimation: Geometry to the Rescue. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision*, pages 740–756, Cham, 2016. Springer International Publishing. 1

[10] James J. Gibson. *The perception of the visual world*. Houghton Mifflin, Oxford, England, 1950. 2

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised Monocular Depth Estimation with Left-Right Consistency. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 7, 8

[12] Junjie Hu, Yan Zhang, and Takayuki Okatani. Visualization of Convolutional Neural Networks for Monocular Depth Estimation. apr 2019. 2

[13] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-Supervised Deep Learning for Monocular Depth Map Prediction. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2215–2223. IEEE, jul 2017. 2, 8

[14] Anthony Lobay and D. A. Forsyth. Shape from Texture without Boundaries. *International Journal of Computer Vision*, 67(1):71–91, apr 2006. 1

[15] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:3061–3070, 2015. 1

[16] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature Visualization. *Distill*, 2017. 2

[17] Teng Leng Ooi, Bing Wu, and Zijiang J. He. Distance determined by the angular declination below the horizon. *Nature*, 414:197–200, 2001. 2

[18] Ruo Zhang, Ping-Sing Tsai, J.E. Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999. 1

[19] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning Depth from Single Monocular Images. *Advances in Neural Information Processing Systems*, 18:1161–1168, 2006. 1

[20] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor Segmentation and Support Inference from RGBD Images. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 1

[21] Rita Sousa, Jeroen B.J. Smeets, and Eli Brenner. Does size matter? *Perception*, 41(12):1532–1534, 2012. 2

[22] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 2

[23] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning Depth From Monocular Videos Using Direct Methods. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 8

[24] Matthew D. Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 818–833, Cham, 2014. Springer International Publishing. 2

[25] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised Learning of Depth and Ego-Motion from Video. In *CVPR*, page 7, 2017. 1, 2, 8