Monocular Depth Map Estimation Based on a Multi-scale Deep Architecture and Curvilinear Saliency Feature Boosting

Saddam Abdulwahab[1*], Hatem A. Rashwan[1†], Miguel Angel Garcia[2†], Armin Masoumian[1†] and Domenec Puig[1†]

[1*]Dept. of Computer Engineering and Mathematics, Universitat Rovira i Virgil, Carretera de Valls, Tarragona, 43007, Tarragona, Spain.

[2]Dept. of Electronic and Communications Technology, Universidad Autnoma de Madrid, Ciudad Universitaria de Cantoblanco, Madrid, 28049, Madrid, Spain.

*Corresponding author(s). E-mail(s): saddam.abdulwahab@urv.cat;
Contributing authors: hatem.abdellatif@urv.cat; miguelangel.garcia@uam.es; armin.masoumian@urv.cat; domenec.puig@urv.cat;
†These authors contributed equally to this work.

Abstract

Estimating depth from a monocular camera is a must for many applications, including scene understanding and reconstruction, robot vision, and self-driving cars. However, generating depth maps from single RGB images is still a challenge as object shapes are to be inferred from intensity images strongly affected by viewpoint changes, texture content and light conditions. Therefore, most current solutions produce blurry approximations of low-resolution depth maps. We propose a novel depth map estimation technique based on an autoencoder network. This network is endowed with a multi-scale architecture and a multi-level depth estimator that preserve high-level information extracted from coarse feature maps as well as detailed local information present in fine feature maps. Curvilinear Saliency (CS), which is related to curvature estimation, is exploited as a loss function to

boost the depth accuracy at object boundaries and raise the performance of the estimated high-resolution depth maps. We evaluate our model on the public NYU Depth v2 and Make3D datasets. The proposed model yields superior performance on both datasets compared to the state-of-the-art, achieving an accuracy of **86%** and showing exceptional performance at the preservation of object boundaries and small 3D structures. The code of the proposed model is publicly available at https://github.com/SaddamAbdulrhman/MDACSFB.

Keywords: Monocular depth map estimation, deep autoencoders, multiscale networks, curvilinear saliency.

1 Introduction

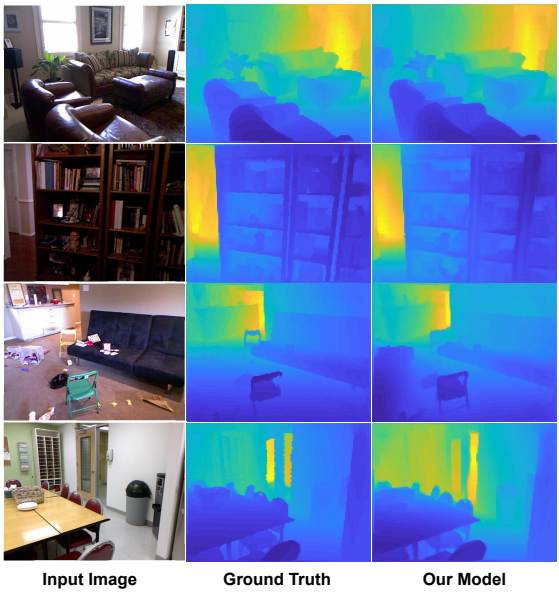

Fig. 1 Comparison of estimated depth maps: input RGB images, ground-truth depth maps, estimated depth maps with the proposed model.

Depth map estimation from a single intensity image is an innately challenging task since multiple 3D shapes can project into a same 2D image. Scene depth estimation plays an essential role in computer vision as it leverages the perception and understanding of natural 3D scenes. This is beneficial for many applications, such as industrial robots [27], self-driving cars [26], augmented reality [28], 3D reconstruction [63], human activity recognition [49], and other

fields. In addition, estimating a depth map from a single image is crucial for determining the 3D pose of the objects present in a scene.

Estimating depth maps (also known as depth images) from a monocular camera is not new. Numerous works have been proposed based on monocular cues extracted from RGB images, such as texture variations and image gradients [15, 16]. Recently, with the outstanding progress of deep learning, several methods based on deep networks have been proposed for 3D shape generation from a single color image of an object [13, 21]. Different deep models are typically used for image-to-image translation in order to learn the mapping among multiple domains, such as Fully Convolutional Networks (FCN) [7], U-Net networks [5], and Generative Adversarial Networks (GAN) [23, 59, 59]. The majority of deep network models for depth map estimation are trained from RGB images and the corresponding depth images captured by range cameras or LiDAR sensors [1, 10].

Feature aggregation is beneficial to generate more accurate depth maps by integrating into a single feature map the response maps obtained at different scales. Various feature aggregation approaches have been proposed, such as the method presented in [25]. It applies a feature pyramid to aggregate multiple-scale features through a fusion network. The latter can integrate the features extracted by several encoder layers through adaptive fusion mechanisms that aggregate coarse depth maps in order to predict fine depth maps. In [57], the authors introduce the side prediction aggregation method for fast monocular depth-map estimation. The proposed network enhances the embedding of scene structural information from low-level to high-level layers. They apply continuous spatial refinement loss at multiple resolutions to improve the accuracy of their prediction model. Besides, the proposed model can further perform adversarial learning at multiple resolutions with minor additional computation. In [61], the authors combine different super resolution methods by applying semantic information in order to build an adaptive group-structured sparse representation approach that makes full use of non-local dependency information of external HR references. Furthermore, the model proposed in [54] addresses the problem of monocular human depth estimation via pose estimation. They use PoseNet and DepthNet to estimate keypoint heat maps and a depth map, respectively. They introduce a feature blending block to make the networks learn to predict depths more accurately by adding the pose information extracted by PoseNet and the features extracted by DepthNet into the next layer of DepthNet.

The present work proposes an autoencoder network, a cutting-edge technique for image-to-image translation, as a baseline network for predicting a depth map from a single color image. Our work is close in spirit to that of [14, 17] in the sense that we also make use of a deep learning approach to estimate depth maps from a single image. The proposed model is based on an autoencoder network with skip connections, a multi-level depth estimator included in the decoder network, and a loss function based on Curvilinear Saliency (CS) [19]. All those components are integrated into a single pipeline

to estimate depth maps from a monocular camera. Our method is promising since it can estimate depth maps for both indoor and outdoor scenarios. In addition, it yields results with a high precision rate and an acceptable computational cost compared to the state-of-the-art. Our results show that the proposed model yields high-resolution depth maps that preserve object boundaries and small details with high accuracy. Fig. 2 shows the proposed depth map estimation framework. The main contributions of this paper are:

- We propose a deep autoencoder for depth estimation based on the SENet-154 network introduced in [47]. Thus, the encoder's backbone is SENet, which integrates Squeeze and Excitation (SE) blocks into the ResNeXt-152 network presented in [50]. The ResNeXt-152 used in this work was defined with cardinality 64 and bottleneck width 4D. SENet helps the autoencoder to exploit the split-transform-merge strategy by aggregating a set of transformations applied to the input features. Moreover, the representational power of the autoencoder is improved by performing dynamic channel-wise feature recalibration through SE blocks.
- We propose the integration of a depth-map predictor at every layer of the decoder network in order to refine the final estimated depth map by preserving global information present in the coarse feature maps as well as detailed local information contained in the fine feature maps. Corresponding feature maps from the encoder are concatenated in the decoder with the up-sampled depth predictions and the deconvolution of the feature maps fed by the previous decoder layers.
- We propose Curvilinear Saliency (CS), a curvature estimator introduced in [19], as a loss function aimed at enhancing depth map edges.

The rest of the paper is organized as follows. Section 2 summarizes the related work. Section 3 details the proposed method to estimate depth maps from single color images. Section 4 describes the network training procedure. Section 5 presents experimental results and the obtained performance. Finally, Section 6 concludes this work, suggesting future research lines.

2 Related Work

This section presents a short review of previous work related to monocular depth map estimation through both classical computer vision and deep learning, autoencoder networks and curvilinear saliency.

2.1 Depth Map Estimation

Depth map estimation from a single RGB image keeps being a very challenging task due to the limited availability of information and inherent ambiguity.

The problem has attracted a lot of attention over the past years, leading to a wide variety of approaches. Many of those solutions are based on classical computer vision. For example, [29] proposes a method for metric depth estimation for UAVs by combining computer vision and odometry

with unsupervised machine learning. In turn, [30] applies traditional Structure from Motion (SfM) in order to reconstruct the 3D structure of the scene and estimate the camera motion from potentially extensive image collections even covering whole cities. These classic methods apply a relatively long sequence of stages. They start with the registration of consecutive images by finding correspondences between geometric features extracted from the images through well-known techniques such as [31]. These methods model hand-crafted features to infer depth information, but those features lack generality across different real-world scenes. Hence, classical approaches have considerable difficulty to yield reasonable accuracy.

Given the significant progress of deep learning, several approaches based on deep networks have successfully been proposed to predict depth maps from single images. For instance, [25] introduced SynSin, an end-to-end model to perform single image view synthesis. The authors used the well-known UNet network model [5], with eight down-sampling and up-sampling layers followed by a sigmoid layer and a renormalization step to yield a final predicted depth map. However, this may fail to preserve the scene's structure accurately. In [11], the authors presented a framework for depth and surface normal estimation from a single image. It consists of a regression stage using a deep CNN model to learn the mapping from multi-scale image patches to depth or surface normal values at the super-pixel level. The SLIC algorithm proposed in [6] was used to obtain the super-pixels. [6] then refined the estimated super-pixel depth or surface normal to the pixel level by exploiting the potentials on the depth or surface normal maps. It considers a data term, a smoothness term among super-pixels and an auto-regression term characterizing the local structure of the estimated depth map. A three-layer CNN network trained with a per-pixel Euclidean loss was presented in [9] to transform the given color image into a geometrically meaningful output image. In addition, this method uses Conditional Random Fields (CRF) as a loss layer to enforce local consistency in the output image.

Recently, by benefitting from the capability to capture context information, the model proposed in [55] applies an end-to-end unsupervised deep learning framework based on an encoder-decoder network for monocular depth-map estimation. That method integrates attention blocks to explore more general contextual information among the feature volumes, as well as a multi-wrap loss function to further improve the original disparity estimation from the network. Alternatively in [58], the authors propose a semi-supervised method that combines the advantages of both supervised and self supervised approaches. That method addresses the problem of monocular depth-map estimation by using a small number of image depth pairs. They apply a generator and two discriminators. The generator network estimates depth whereas the two discriminator networks inspect the estimated depth-image pair and depth, respectively. Although the detection performance of salient objects from a single color image is improved, it is still challenging to yield satisfactory results

for images with cluttered backgrounds. Unfortunately, semi-supervised training does not always guarantee good performance, as these networks are unable to correct their bias and require additional domain information, such as camera focal length and sensor data. In [60], a novel regularizer loss function for monocular depth-map estimation is proposed. It is adaptively learned by a tiny CNN Regularizer Net in an adversarial way. It could further replace the hand-crafted gradient loss and normal loss functions. Although the method preserves far richer geometric details and more accurate object boundaries, it still requires a long time to converge and sometimes presents instability problems during the adversarial training process. In our previous work [36], we proposed a deep learning model to estimate a depth map of an object depicted in a single image. That map is then used for predicting the 3D pose of the object. The proposed model consists of two subsequent autoecoder networks based on a Generative Adversarial Neural network (GAN). The main disadvantage of this model is that it assumes a cross-domain training procedure for 3D CAD models of objects appearing in real photographs, not for the complete scene.

In turn, [39] developed a deep ordinal regression network for monocular depth estimation by training the network with an ordinary regression loss. A multi-scale network structure was adopted to avoid unnecessary spatial pooling and capture multi-scale information in parallel. However, this method produces sharp discontinuities in the object shapes. In [40], the authors proposed a method for monocular depth map estimation based on two stages: a dense feature extractor and a depth map generator. The first stage extracts features from the input image while keeping dense feature maps. An attention mechanism was integrated into the depth map generator to fuse multi-scale features maps. Although this model can preserve the structural details of the scene depth, it still lacks precision for complex objects. Finally, new proposals have emerged for depth map estimation from a single image based on CNNs [8, 14]. In particular, [8] introduced a residual network to solve the problem of estimating the depth map from a given single RGB image. They also introduced the reverse Huber loss and newly designed up-sampling modules. The model is composed of a single architecture trained end-to-end.

The aforementioned deep learning approaches have been proven to yield the most accurate results. In this line, we propose a method based on a deep network model for estimating depth maps from single color images. Our model differs from previous work in which it successfully keeps the scene's structure for both indoor and outdoor scenarios, showing significant performance in the preservation of the boundaries and small structures of objects.

2.2 Autoencoder Networks

Autoencoders play a fundamental role in deep learning for image-to-image translation and other related tasks. They learn to map data from a domain A to a domain B. These models are usually trained by minimizing a reconstruction loss function that measures the difference between the reconstructed

output and its ground-truth. Recently, autoencoders have been applied to many vision-related problems, such as image reconstruction [32], image registration [33], image segmentation [34], Human health posture [56]. Thus, they are also advantageous for depth map estimation. In addition, they have been used with great success for both supervised and unsupervised tasks, such as [35–38]. The main advantage of autoencoders is that they provide a deep model directly based on the input data rather than on predefined filters. Besides, they reduce the dimensionality of the data used for training.

We apply an autoencoder network for depth map estimation as shown in Fig. 2. It is based on the SE-ResNet model (Fig. 3) to capture latent spatial structures of the input images for both the training and inference models.

2.3 Curvilinear Saliency

A depth map is an image that represents information about the distance between the 3D surfaces present in a scene and the camera. The quality of a depth map must be assessed based on geometrical cues extracted from it. Most approaches [48, 51] compare the gradients of their estimated depth maps with the ground-truth through a loss function in order to train their deep models. However, using such gradients as a quality measure is not accurate enough [18, 20, 22]. Indeed, it is essential to detect valleys and ridges related to curvature measurements where the camera and the light source are in the same (or opposite) direction. Those features have the advantage of representing both outer and inner (self-occluding) contours of the scene objects, which are useful for estimating the pose and viewpoint.

Consequently, robust valley and ridge detectors can improve the training process of deep models aimed at depth map estimation. In previous work, we proposed the Curvilinear Saliency (CS) detector [20, 22] for extracting the surface discontinuities of the objects in a scene. It extracts geometrical features that are robust to light and viewpoint changes. We apply CS features through a loss function in order to improve the network's performance by boosting the depth estimation accuracy under the extrinsic characteristics associated with the color image acquisition, such as the camera pose and light conditions.

3 Proposed Method

This section describes the main stages of the proposed method to estimate a depth map from a single RGB image, as well as the tools and resources used in this work. Fig. 2 shows an overview of the proposed network model. Its main component is an autoencoder network with skip connections that applies a multi-level depth predictor in the decoder. The performance of the autoencoder is improved by applying a loss function based on CS features. We formulate the problem in subsection A. In the remaining subsections, we detail the proposed method.

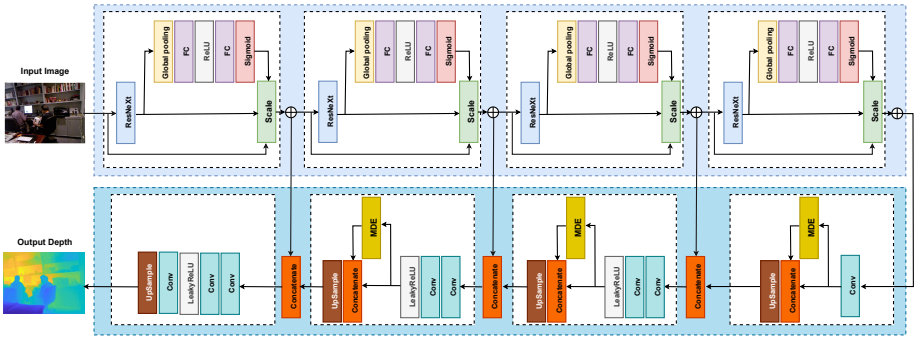

Fig. 2 Overview of the proposed deep network model.

3.1 Problem Formulation

Let $A \in \mathbb{A}$ be a 2D color image. The problem of generating its corresponding depth map, $B \in \mathbb{B}$, can be formally stated as the definition of a function $f : \mathbb{A} \rightarrow \mathbb{B}$ that maps elements from domain $\mathbb{A}$ to elements in its co-domain $\mathbb{B}$. We introduce an efficient deep learning-based system for depth map estimation from a single RGB image. Specifically, we propose an autoencoder network that consists of two consecutive networks: an encoder and a decoder. The decoder D estimates a depth map $\hat{B}$ from the latent representation generated by the encoder E when applied to the given color image A: $\hat{B} = D(E(A))$. A loss function $CS(B, \hat{B})$ is used to compare the estimated depth map $\hat{B}$ with the ground-truth B. The next subsections describe the architecture of our proposed system in detail.

3.2 Network Architecture

Fig. 2 shows an overview of our autoencoder network for depth map estimation. It is composed of an encoder and a decoder. The encoder is fed with an RGB image and transforms it into a latent representation of high-level features. The decoder then maps that latent representation to a depth map.

3.2.1 Encoder

Inspired by [14], the input RGB image is encoded into a latent representation by applying the first four blocks of the SENet-154 network [47] pre-trained on ImageNet [41]. SENet-154 applies a multi-scale and multi-crop fusion strategy for extracting rich high-level features from the input images. It integrates Squeeze-and-Excitation (SE) blocks into a modified version of ResNeXt-152, which is an extension of the ResNeXt-101 model by following the block stacking of ResNet-152. Fig. 3 shows the structure of a single SE block integrated into the ResNet residual block. ResNeXt derives from ResNet by aggregating the output of multiple bottleneck residual blocks defined in a low-dimensional

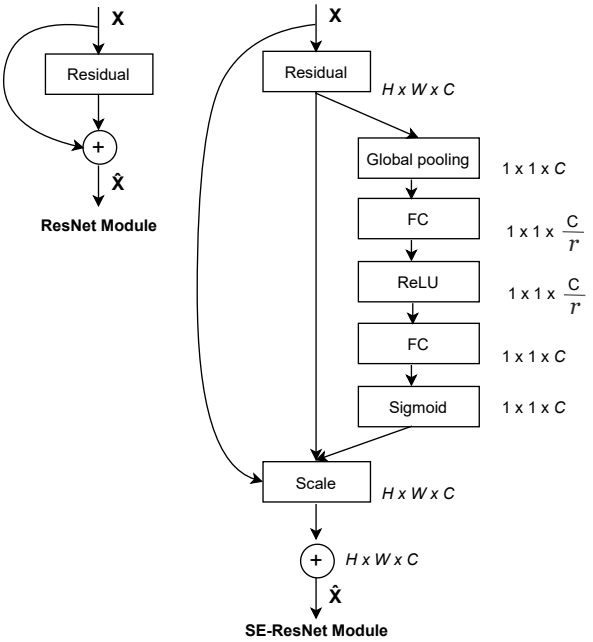

Fig. 3 Scheme of SE-ResNet modules [47]. Reduction ratio r set to 16.

embedding (1×1 and 3 × 3 convolutions are applied to 4 instead of 64 channels), as shown in Fig. 4. The main parameters of ResNeXt are 1) the number of aggregated residual blocks, referred to as cardinality, and 2) the number of channels processed in each residual block, referred to as depth (see Fig. 4). In this work, we set cardinality to 64 and depth to 4. Higher cardinality yields a more accurate representation of the input images and raises accuracy, as explained in [47].

The proposed encoder is fed with input RGB images of 480×360 (*width* × *height*) pixels (see Fig. 2). Its first convolutional block generates 128 feature maps (channels) of size 240×180. In turn, the second block outputs 256 feature maps of size 120×90. The third block generates 512 feature maps of size 60×45. Finally, the last block gives 1024 coarse-level feature maps of size 30×23, which constitute the encoder's latent representation.

3.2.2 Decoder

The decoder network consists of four convolutional blocks as shown in Fig. 2. The first block applies a 3×3 convolution with stride 1 to the channels generated by the encoder network in order to project the high-level features extracted by the encoder across channels. The resulting feature map is fed into a Multi-level Depth Map Estimator (MDE) described in the following subsection, which predicts a coarse depth map of size $23 \times 23 \times 1$. That depth map

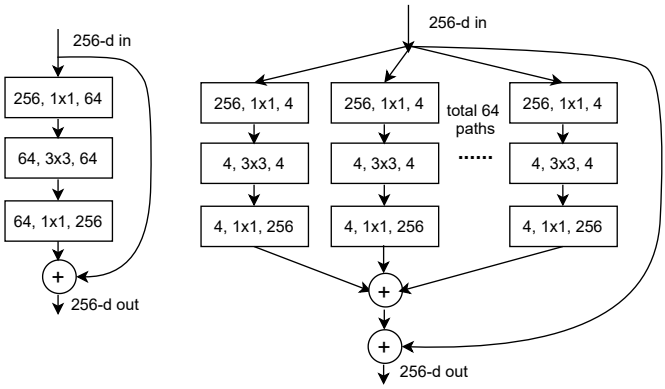

Fig. 4 Left: bottleneck residual block of ResNet [52]. Right: residual block of ResNeXt with cardinality 64, depth 4, and roughly the same complexity [47]. Every layer is depicted as (# in channels, filter size, # out channels).

is concatenated with the feature map generated by the initial convolution. The result is upsampled to the spatial resolution of the next decoder's block through 2×2 bilinear upsampling [42]. The upsampled feature map is concatenated with the output features of the corresponding block from the encoder (skip connection) before feeding it to the next convolutional block.

The next two convolutional blocks apply two consecutive 3×3 convolutions with output channels set to half the number of input channels in order to improve the representation of the input feature map. A LeakyReLU activation function [12] with $\alpha = 0.2$ is applied to the output of the second convolution for speeding up the training process. The feature map generated by every activation function is concatenated with the output of its corresponding MDE layer to predict a finer multi-scale depth map. The resulting feature map is rescaled using 2×2 bilinear upsampling and then concatenated with the features from the corresponding encoder block.

The last convolutional block of the decoder generates the final depth map. Similarly to the previous decoder's blocks, it consists of two consecutive 3×3 convolutions with output channels set to half the number of input channels, followed by a LeakyReLU with $\alpha = 0.2$. A 1×1 convolution is applied for adapting the filter space dimensionality to the size of the required depth maps. A 2×2 bilinear upsampling is then applied for upscaling the feature maps. The output of the decoder network is a depth map of size 240×180 for NYU Depth-v2 and 86×115 for Make3D.

3.2.3 Multi-level Depth Map Estimator

In order to learn the scale-aware depth map context by leveraging context-aware spatial features extracted at different scales, Multi-level Depth map Estimators (MDEs) are applied within the decoder as shown in Fig. 2. MDEs help preserve object structure detail and thus yield crisp boundaries, especially

in complex environments. In particular, an MDE layer is included in the first three convolutional blocks of the decoder. The MDE in the first decoder's block is fed with the output of its 1×1 convolutional layer, whereas the next two MDEs are fed with the result of their respective LeakyReLU functions. An MDE consists of a 1×1 convolution with a single channel followed by a ReLU activation function. The output of every MDE is concatenated with its input feature map and then rescaled through 2×2 bilinear upsampling prior to feeding the result into the next decoder's block.

4 Network Training

The majority of depth-map estimation methods compare the depth maps they generate with their corresponding ground-truth by means of differentiation operators that approximate the local 2D gradients, such as the Sobel filter. Alternatively, we propose the use of the Curvilinear Saliency (CS) described in the previous section in order to highlight the geometry of objects with disregard of texture and light changes. In particular, the proposed autoencoder has been trained by aggregating two loss functions: the CS loss and the content loss. The CS loss accounts for the dissimilarity between the curvilinear features of both the estimated B and real (ground-truth) $\hat{B}$ depth maps. In turn, the content loss follows a classical approach in which the estimated depth maps are compared with their corresponding ground-truth in an element-wise fashion.

4.1 Curvilinear Saliency Loss

The proposed CS loss function compares the curvilinear saliency of both the estimated and ground-truth depth maps. CS features [19] allow us to approximate the curvatures of depth maps, being able to assess the quality of the generated estimations in terms of representation fidelity of surface edges and discontinuities. The features extracted by CS have several advantages, especially when extracting the local structure of the points of interest. In addition, these features are invariant to viewpoint changes and transformations that do not change the shape of the surface. CS depends on the principal curvatures, which are decisive parameters that fully describe a local surface shape. CS provides a unified way of treating ellipses and hyperbolas with real conics, concave, convex, saddle-shaped and parabolic. The CS loss thus behaves as an edge-aware error function.

A depth map (also known as depth image) $B(x, y)$ associates every element (x, y) with a z-coordinate (depth) related to the distance from a certain 3D surface point to the camera coordinate frame. Let $\mathbb{D}$ be the 3D surface represented in $B(x, y)$. Every 3D point $D \in \mathbb{D}$ can be defined as: $\mathbb{D} = [x, y, B(x, y)]$. CS aims at detecting local surface discontinuities by means of the maximum principal curvature (κ_1) in one direction and the minimum principal curvature (κ_2) in the orthogonal direction. CS uses the difference between both principal curvatures ($\kappa_1 - \kappa_2$) to represent the ridges and valleys present in depth

maps. Let $\hat{N}(x, y)$ be the unit normal vector of $\mathbb{D}$ at point D:

$$\hat{N} = D_x \times D_y = \alpha \begin{bmatrix} \nabla B \\ 1 \end{bmatrix},$$

where the gradient of B at D is $\nabla B = [B_x, B_y]^T$, and $\alpha = 1/\sqrt{1 + \nabla B^2}$. Since the two columns of the Jacobian matrix J_D of $\mathbb{D}$ are $D_x = [1, 0, B_x]^T$, and $D_y = [0, 1, B_y]^T$, the first fundamental form of $\mathbb{D}$ at D can be computed as:

$$\mathrm{I}_D = I_{2\times 2} + \nabla B \nabla B^T,$$

where $I_{2\times 2}$ is the 2×2 identity matrix.

In turn, the second fundamental form of $\mathbb{D}$ at D can be obtained as:

$$\mathrm{II}_D = \alpha H_B,$$

where H_B is the Hessian of B, which represents the second-order partial derivatives of B along the x and y directions.

As explained in [19], the principal curvatures of $\mathbb{D}$ at D, $\{\kappa_1, \kappa_2\}$, correspond to the eigenvalues of $M = \mathrm{I}_D^{-1}\mathrm{II}_D$:

$$M = \begin{bmatrix} (B_y^2 + 1)B_{xx} - B_x B_y B_{xy} & (B_y^2 + 1)B_{xy} - B_x B_y B_{yy} \\ (B_x^2 + 1)B_{xy} - B_x B_y B_{xx} & (B_x^2 + 1)B_{yy} - B_x B_y B_{xy} \end{bmatrix}.$$

Let λ_1 and λ_2 be the eigenvalues of M obtained as:

$$\lambda_\pm = \frac{1}{2}[-trace(M) \pm \sqrt{trace^2(M) + 4\det(M)}],$$

where *trace* is the sum of elements in the main diagonal of M, det is the determinant of M, and $\lambda_1 = \lambda_+$, $\lambda_2 = \lambda_-$. Finally, CS is defined as:

$$CS = \kappa_1 - \kappa_2 = (\lambda_1 - \lambda_2)\nabla B.$$

For every depth map we can generate a CS image as shown in Fig. 5. The CS loss function between the estimated depth map $\hat{B}$ and its ground-truth B is defined as the mean squared error of their respective CS images:

$$L_{CS}(B, \hat{B}) = \frac{1}{wh}\sum_{x=1}^{w}\sum_{y=1}^{h}[CS_B(x, y) - CS_{\hat{B}}(x, y)]^2,$$

where w and h is the width and height of the depth maps, respectively.

4.2 Content Loss

The content loss measures the similarity between the shape of the estimated depth map $\hat{B}$ and its ground-truth B by means of three separate loss functions

Fig. 5 Color images (Row 1), associated depth images (Row 2) and their corresponding CS images (Row 3).

that are added together. The first loss function is the point-wise L1-norm defined on the depth values:

$$L_{L1}(B, \hat{B}) = \frac{1}{wh} \sum_{x=1}^{w} \sum_{y=1}^{h} |B(x, y) - \hat{B}(x, y)|.$$

The second loss function is the structural similarity index measure (SSIM). It is a method for predicting the perceived quality of digital images by measuring the similarity between them. In this case, the SSIM index is computed between B and $\hat{B}$:

$$L_{SSIM}(B, \hat{B}) = \frac{1 - \frac{(2\mu_{\hat{B}}\mu_B + c_1)(2\sigma_{\hat{B}B} + c_2)}{(\mu_{\hat{B}}^2 + \mu_B^2 + c_1)(\sigma_{\hat{B}}^2 + \sigma_B^2 + c_2)}}{2},$$

where $\mu_{\hat{B}}$ and $\sigma_{\hat{B}}$ are the mean and standard deviation of $\hat{B}$, respectively, μ_B and σ_{μ_B} are the mean and standard deviation of B, respectively, $\sigma_{\hat{B}B}$ is the covariance of $\hat{B}$, $c1 = 0.01^2$ and $c2 = 0.03^2$.

The third loss function is the Mean Squared Error (MSE) between B and $\hat{B}$:

$$L_{MSE}(B, \hat{B}) = \sum_{x=1}^{w} \sum_{y=1}^{h} \frac{(B(x, y) - \hat{B}(x, y))^2}{wh}.$$

4.3 Final Objective Loss

The final training loss $L(B, \hat{B})$ of the proposed autoencoder is defined as a weighted average of the CS loss and the three loss functions that define the

content loss:

$$L(B, \hat{B}) = \lambda L_{CS}(B, \hat{B}) + (1 - \lambda)(L_{L1}(B, \hat{B}) + L_{SSIM}(B, \hat{B}) + L_{MSE}(B, \hat{B})),$$

where λ is a weighting factor set to 0.5 in this work.

5 Experiments and Results

This section describes the experiments performed to evaluate the proposed model, in addition to the dataset and evaluation measures used in the experiments.

5.1 Datasets

We conducted all the experiments in this work on two publicly available datasets: NYU Depth-v2 [24] for indoor scenes and Make3D [2] for outdoor scenes.

5.1.1 INDOOR SCENES

NYU Depth-v2 is a public dataset that provides color images and depth maps for different indoor scenes captured at a resolution of 640×480 pixels [24]. The dataset contains raw frames captured by scanning various indoor scenes with a Microsoft Kinect: 120K frames for training and 654 for testing [10]. We trained our network model on a subset of Depth-v2 containing $50,000$ images as proposed in [14]. We resized all color images from 640×480 to 480×360 to feed the network. The depth maps have an upper bound of 10 meters. Fig. 6 shows some examples from NYU Depth-v2.

5.1.2 OUTDOOR SCENES

Make3D is a public outdoor dataset [2] with 400 training and 134 test images captured through a custom-built 3D scanner. The resolution of the ground-truth depth map is limited to 305×55 pixels, whereas the original size of the RGB images is $2,272 \times 1,704$ pixels. To increase the number of training samples, we applied the data augmentation techniques described in the next subsection. We extended the original 400 training images to $11,000$ images. Increasing the number of training images allowed the developed depth-map estimation model to become more robust. Moreover, we resized all images to 460×345 to feed the network. Fig. 9 shows examples from Make3D.

5.2 Data Augmentation

We applied the following data augmentation techniques to the images contained in the Make3D dataset to increase the number of training samples under different conditions and hence increase the diversity of the training dataset:

- Scale: Every input image and its corresponding depth map were randomly scaled by S ∈ [0.5,1.7].
- Rotation: Every input image and its corresponding depth map were rotated by R ∈ [-60,-45,-30,30,45,60] degrees.
- Gamma Correction: The gamma correction of each input RGB image was randomly varied by G ∈ [1,2.8].
- Flipping: Every input image and its corresponding depth map were flipped by F ∈ [-1,0,1].
- Translation: Every input image and its corresponding depth map were translated by T ∈ [-6,-4,-2,2,4,6] pixels.

Although the represented scenes were slightly warped after applying those data augmentation techniques, we observed that the efficiency of the network significantly improved compared to the model trained without data augmentation.

5.3 Parameter settings

Our network model was trained by applying the Adam optimizer [4] with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and an initial learning rate of 0.0001. The latter was reduced by 10% every 3 epochs for the NYU Depth-v2 dataset. For Make3D, we did not reduce the learning rate during training. The best accuracy was attained after 15 epochs. All experiments were run on a 64-bit Core I7-6700, 3.40GHz CPU with 16GB of RAM and an NVIDIA GTX 1080 GPU on Ubuntu 16.04 and the PyTorch deep learning framework [3]. The training process of the proposed model took around 3 hours per epoch with a batch-size of 2 for NYU Depth-v2, and around 45 minutes per epoch with a batch-size of 4 for Make3D. In turn, the online estimation of depth maps during testing run at around 20,6 milliseconds per image for NYU Depth-v2, and around 35 milliseconds per image for Make3D.

5.4 Evaluation Measures

The performance of the proposed model was evaluated by computing the error between the depth values of the estimated depth map $\hat{B}$ and its ground-truth B. The threshold accuracy measure from [9] is essentially the expectation that the depth value error of a given pixel in $\hat{B}$ is lower than a threshold thr^Z. It is an indication of how often the estimated depth map is correct:

$$\delta_Z = \mathbb{E}_B[\, F(\max(\frac{B(x,y)}{\hat{B}(x,y)}, \frac{\hat{B}(x,y)}{B(x,y)}) < thr^Z)] \ ,$$

where $F(\cdot)$ is an indicator function that yields 1 if the condition in its argument is satisfied and 0 otherwise. Similarly to [9], we set $thr = 1.25$, and $Z \in \{1, 2, 3\}$.

From a quantitative point of view, the final performance of the proposed model was assessed through three commonly-used error measures: the root

mean squared error (rms), which provides a quantitative measure of per-pixel error, the average relative error (rel), and the average log_{10} error:

$$rms = \sqrt{\frac{1}{wh}\sum_{x=1}^{w}\sum_{y=1}^{h}(B(x,y) - \hat{B}(x,y))^2},$$

$$rel = \frac{1}{wh}\sum_{x=1}^{w}\sum_{y=1}^{h}\frac{|B(x,y) - \hat{B}(x,y)|}{B(x,y)},$$

$$log_{10} = \frac{1}{wh}\sum_{x=1}^{w}\sum_{y=1}^{h}|log_{10}B(x,y) - log_{10}\hat{B}(x,y)|.$$

5.5 Results and Discussion

5.5.1 Ablation Study

Firstly, we performed an ablation study in order to assess the impact of different stages of the proposed autoencoder. The following configurations were considered:

- (Baseline: BL) Basic autoencoder with three content loss functions: pointwise $L1$ loss (L_{L1}), mean squared error loss (L_{MSE}), and structural similarity index measure loss (L_{SSIM}).
- (BLSC) BL model with skip connections from the encoder layers to the corresponding decoder layers.
- (BLSC+MDE) BLSC model with multi-scale depth-map estimator.
- (BLSC+CS) BLSC model with CS loss.
- (BLSC+MDE+CS) BLSC model with multi-scale depth-map estimator and CS loss.

Table 1 Quantitative results of the ablation study for depth-map estimation from color images with the NYU Depth-v2 dataset for different evaluation measures: BL, BLSC, BLSC+MDE, BLSC+CS, and BLSC+MDE+CS configurations. Accuracy: higher is better and Error: lower is better.

Method	δ_Z <1.25 ↑	δ_Z <1.25^2 ↑	δ_Z <1.25^3 ↑	rel ↓	rms ↓	log10 ↓
BL	0.833	0.969	0.9928	0.14	0.532	0.056
BLSC	0.842	0.971	0.9931	0.128	0.525	0.054
BLSC+MDE	0.854	0.97	0.991	0.123	0.538	0.531
BLSC+CS	0.8531	0.973	0.993	0.123	0.529	0.527
BLSC+MDE+CS	**0.8591**	**0.973**	**0.9932**	**0.119**	**0.52**	**0.051**

Table 1 shows quantitative results of the ablation study for NYU Depth-v2. The performance of the proposed model (BLSC+MDE+CS) yielded the best results among other variations of the proposed model in terms of δ_Z, as well as rms, rel, and log_{10} errors. The accuracy of $\delta_Z(thr = 1.25)$ improved by around

2.5% compared to the baseline model (BL). As for the *rel* error, the proposed model yielded a significant improvement of 0.021 compared to BL. Adding multi-scale depth-map estimation (MDE) to the baseline model improved the accuracy by 2.1% and reduced the *rel* error by 12%. In turn, applying CS loss also yielded a significant accuracy improvement and a considerable reduction in the *rel* error compared to BL, with 2% and 12% differences, respectively.

Table 2 Quantitative results of the ablation study for different configurations on Make3D. Error: lower is better.

Method	rel ↓	rms ↓	log10 ↓
BL	0.254	7.11	0.126
BLSC	0.212	6.85	0.117
BLSC+MDE	0.207	6.76	0.107
BLSC+CS	0.201	6.71	0.104
BLSC+MDE+CS	**0.195**	**6.522**	**0.091**

Table 2 shows quantitative results of the same ablation study for Make3D. The proposed model BLSC+MDE+CS yielded the lowest errors among the other tested configurations in terms of the rms, rel, and log_{10} errors.

5.5.2 Performance Analysis

Secondly, we compared the proposed model against six alternative models from the state-of-the-art [8, 14, 39, 40, 43, 44]. In Table 3, we show evaluation measures on NYU Depth-v2 for the seven tested approaches. The accuracy of our proposed model was superior for $\delta_Z(thr = 1.25)$, $\delta_Z(thr = 1.25^2)$ and the log_{10} error. $\delta_Z(thr = 1.25)$ shows an improvement of 0.5% compared to [8], the best second method. With respect to $\delta_Z(thr = 1.25^2)$, our model and [14] yielded an improvement of 1% compared to the other five methods. The model proposed in [14] gave the best accuracy for both $\delta_Z(thr = 1.25^3)$ and rms, but with a difference against our proposed model of just 0.0004% and 0.055%, respectively. However, we can note that our model provided the best accuracy for $\delta_Z(thr = 1.25)$, which is the most restrictive threshold. In addition, our model scored the second lowest log_{10} error (0.119), only behind the model proposed in [39], which had the best *rel* error with a difference of only 0.004%. However, the proposed model outperformed the model in [39] in terms of the other four evaluation measures.

In Fig. 6, we show qualitative results on the NYU Depth-v2 dataset for the proposed model (BLSC+MDE+CS) and two state-of-the-art monocular depth-map estimation methods introduced in [14] and [43]. Our model is able to estimate more accurate depth maps that are very close to the ground-truth and that preserve the small details of the depicted objects. In fact, our model preserves the outline of the objects present in the scenes in such a way that those objects can be directly recognized from the depth maps. In contrast, object outlines appear crumbled in the depth maps generated by the other methods.

Table 3 Results for depth-map estimation from color images with the NYU Depth v2 dataset for different measures and state-of-the-art methods. The last row shows results obtained with our proposed model. Accuracy: higher is better and Error: lower is better.

Method	δ <1.25 ↑	δ <1.25^2 ↑	δ <1.25^3 ↑	rel ↓	rms ↓	log10 ↓
Fu et al. [39]	0.828	0.965	0.992	**0.115**	0.509	**0.051**
Laina et al. [8]	0.853	0.965	0.991	0.121	0.592	0.052
Hao et al. [40]	0.841	0.966	0.991	0.127	0.555	0.053
Ramamonjisoa et al. [43]	0.8451	0.9681	0.9917	0.1258	0.551	0.054
Alhashim et al. [14]	0.846	**0.974**	**0.994**	0.123	**0.465**	0.053
Tang et al. [44]	0.826	0.963	0.992	0.132	0.579	0.056
Our model	**0.8591**	**0.9733**	0.9932	0.119	0.52	**0.051**

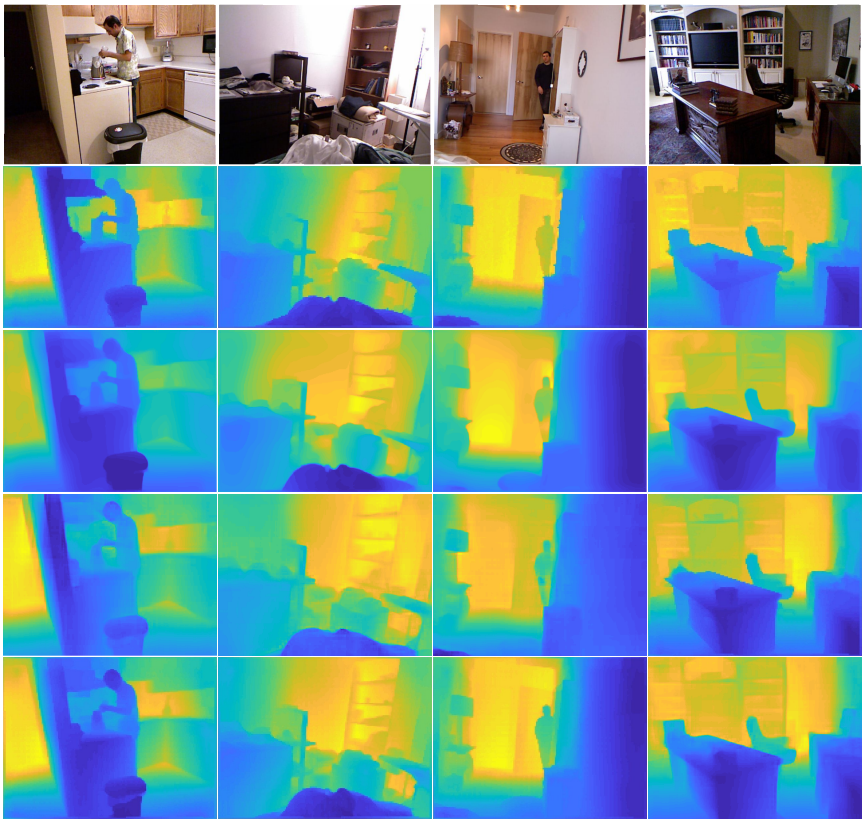

Fig. 6 Input images, ground-truth depth maps and estimated depth maps with the NYU Depth-v2 dataset: color images (Row 1), ground-truth depth maps (Row 2), depth maps generated by Alhashim et al. [14] (Row 3), depth maps generated by Ramamonjisoa et al. [43] (Row 4), and depth maps generated by our model (BLSC+MDE+CS) (Row 5).

One of the main strengths of the proposed method is to use CS as a feature extractor, as it is based on the principal curvatures. CS is responsible for increasing the ability of the model to learn under different conditions, such as (distance, illumination, and colour). Thanks to CS, the model learned

the correct cardinality (i.e., object boundaries) inside the images. Of course, no trained model will generate results better than the ground truth that it attempts to mimic. The trained model can learn from different examples in the dataset, including correct examples of the objects, to improve its performance. For instance, with the NYU Depth v2 dataset, Fig. 7 shows some of the correct examples that intervene in the training process: column 1 shows the objects that are close to the camera, column 2 shows the objects that are far away from the camera, column 3 shows the objects affected by strong illumination, and column 4 shows the objects whose colour is similar to the one of the background. Based on these examples, our model can learn to predict depth even with noisy ground truth in some examples.

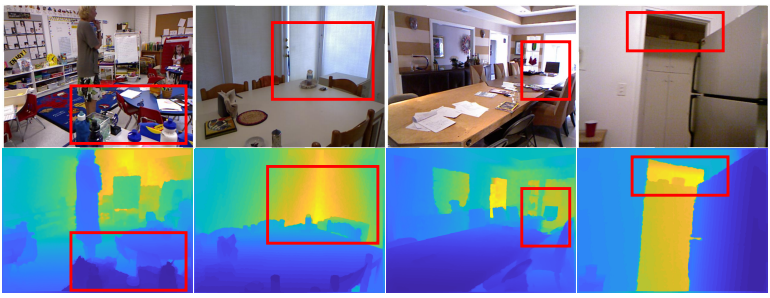

Fig. 7 Some correct examples of the NYU Depth v2 dataset under different conditions: (Column 1) objects that are close to the camera, (Column 2) objects that are far away from the camera, (Column 3) objects affected by strong illumination, and (Column 4) objects whose color is similar to the one of the background.

To assess the overall improvement on the NYU Depth-v2 dataset, in Fig. 8, we show some examples that contain geometrically rich areas. The red box shows the selected geometrically rich areas of the scene and the corresponding estimated depth images. As expected, our depth-map estimation model is able to predict accurate depth with sharp object boundaries. In addition, in order to show quantitative results, we compute the evaluation measures (rel, rms, $log10$, $Accuracy_{\delta}$) for the examples of rich areas shown in Fig. 8, as shown in 4. Notably, these results support the ones presented in Table 3.

Table 4 Results of the four selected geometrically rich areas shown in Fig. Accuracy: higher is better and Error: lower is better. 8.

#	δ_Z <1.25 ↑	δ_Z <1.25^2 ↑	δ_Z <1.25^3 ↑	rel ↓	rms ↓	log10 ↓
1	0.9098	0.9548	0.9754	0.121	0.527	0.0525
2	0.8372	0.9169	0.9645	0.132	0.543	0.0537
3	0.9405	0.9950	0.9990	0.116	0.523	0.0521
4	0.8980	0.9354	0.9698	0.129	0.532	0.0528
Average	0.896375	0.950525	0.977175	0.1245	0.53125	0.052775

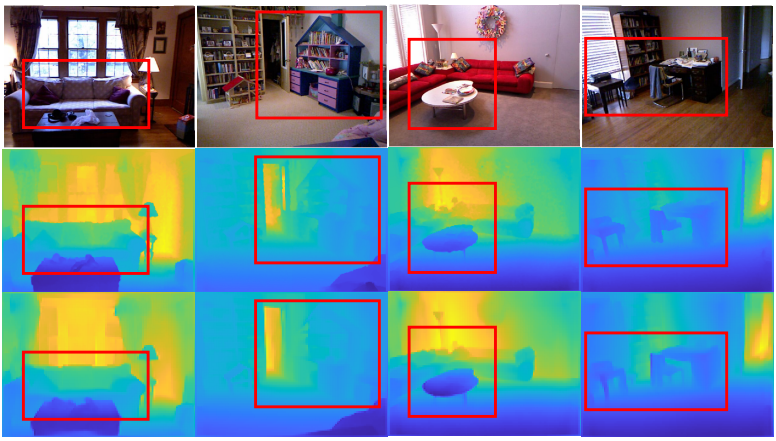

Fig. 8 Exmaples of geometrically rich areas selected from f the NYU Depth v2 test set: (Row 1) shows the original Image, (Row 2) shows the ground-truth depth maps, (Row 3) shows the estimated depth Image.

As for the Make3D dataset, we also compared the proposed model with six alternative methods [9, 11, 44–46, 48]. Table 5 shows the obtained evaluation measures for the seven tested methods. In this case, the proposed model performed similarly to the alternative models. However, our model gave the lowest error for both *rel* and *rms*: *rel* shows an improvement of 0.081% with respect to the other methods, whereas *rms* shows a significant improvement of 0.468%. However, the model proposed in [44] had the lowest error for log_{10}, although with an insignificant improvement of 0.005% with respect to our model. As a conclusion, our model outperformed the tested models with significant improvements or achieved very similar results on the two datasets.

Table 5 Results for depth-map estimation from color images with the Make3D dataset for different measures and state-of-the-art methods. The last row shows results obtained with our proposed model. Error: lower is better.

Method	rel ↓	rms ↓	log10 ↓
Kevin et al. [53]	0.361	15.1	0.148
Godard et al. [48]	0.443	11.513	0.156
Liu et al. [9]	0.314	8.60	0.119
Liu et al. [11]	0.278	7.19	0.092
Kuznietsov et al. [46]	0.421	8.24	0.190
Tang et al. [44]	0.276	6.99	**0.086**
Our model	**0.195**	**6.522**	0.091

For a qualitative assessment on the Make3D dataset, Fig. 9 shows the depth maps estimated from monocular color images by our proposed model and other state-of-the-art methods, such as [9] and [53]. The depth maps generated by our model are depicted in Row 5. The four examples shown in Fig. 9 agree with the results obtained for the NYU Depth-v2 dataset (Fig. 6). Indeed, the

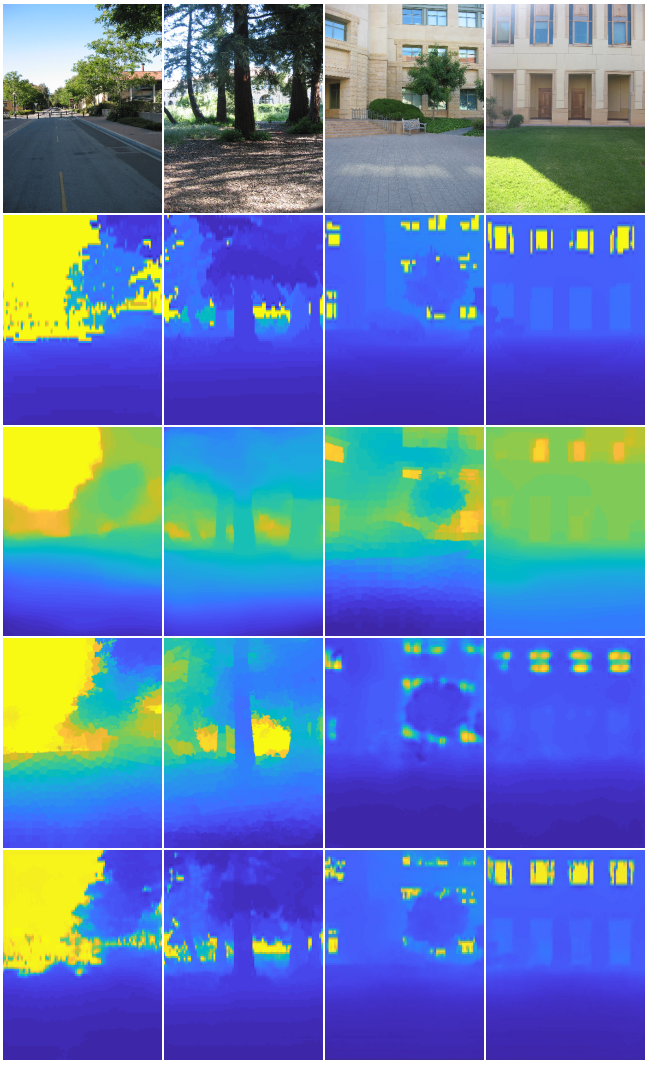

Fig. 9 Input images, ground-truth depth maps and estimated depth maps with the Make3D dataset: color images (Row 1), ground-truth depth maps (Row 2), depth maps generated by Liu et al. [9] (Row 3), depth maps generated by Kevin et al. [53] (Row 4), and depth maps generated by our model (Row 5).

proposed model can estimate more accurate depth maps than the other tested models for outdoor scenes even under different illumination conditions.

To further assess the performance of the proposed model, we randomly selected images from the NYU Depth-v2 and Make3D test subsets in order to show the ability of the proposed model to estimate accurate depth maps (see Fig. 10 and Fig. 11). Notice that our model can produce accurate results with high-quality depth maps. For instance, regarding NYU Depth-v2, Fig.

10 shows that our model can generate depth maps not only better than the other methods, but also capturing some details that are not even present in the ground-truth. For instance, the example shown in Fig. 10-(Column 3) depicts some baskets on the floor that appear blurred in the ground-truth. However, they are shown in detail in the depth maps generated by our model. In general, our model can estimate correct depth values for objects that are close to the camera (see Column 1), for objects that are far away from the camera (see Column 2), as well as for objects affected by strong illumination (see Column3). The model can also detect the boundaries between objects whose color is similar to the one of the background (see Column4).

Additional results with the Make3D dataset shown in Fig. 11 indicate that our model can estimate correct depth values for buildings that are far away from the camera (see Column1), as well as for trees that are close to the camera (see Column 2). Moreover, the model is robust to shadows (see Column 3) and distinguishes objects that have the same color and are close to each other (see Column 4).

All in all, the depth maps generated by our proposed model (BLSC+MDE+CS) keep the boundaries and details of the objects present in the scene. That preservation of shape discontinuities is likely to be beneficial for generating more accurate semantic maps and for improving the visual odometry of autonomous vehicles. Furthermore, the previous results show that our model can be trained even with noisy ground-truth depth maps. Another remarkable point is the fact that the proposed model achieves these promising results without applying any refinement steps.

6 Conclusion

We have introduced an efficient deep network model for estimating a high-resolution depth map from a single color image. The proposed model is based on an autoencoder network with skip connections between the corresponding layers of its encoder and decoder branches. For estimating accurate depth maps, we have proposed the introduction of multi-scale depth-map estimation layers in the decoder branch. Moreover, the application of the Curvilinear Saliency (CS) as a loss function during the training process has also been proposed to enhance depth-map edges.

The performance of the proposed model has been evaluated on the NYU Depth-v2 and Make3D datasets, obtaining promising results with a high precision rate and an acceptable computational cost. These results show that it is feasible to estimate complex depth maps from monocular color images. Future work aims at applying the proposed model to estimate object volumes based on a single RGB camera.

Acknowledgments. Financial support was given by the pre-doctoral grant (FI 2020) funded by the Catalan government.

All authors declare that they have no conflicts of interest.

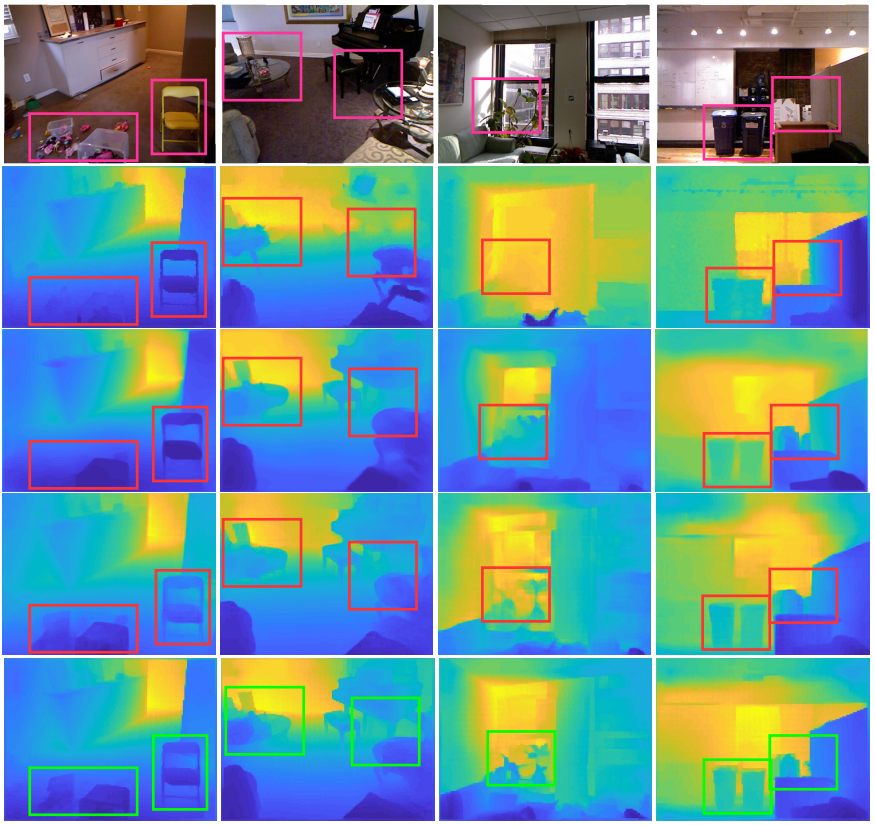

Fig. 10 Qualitative analysis of the proposed model with the NYU Depth-v2 dataset: color images (Row 1), ground-truth depth maps (Row 2), depth maps estimated by Alhashim et al. [14] (Row 3), depth maps estimated by Ramamonjisoa et al. [43] (Row 4), and depth maps estimated by our model (Row 5).

References

[1] Ge L, Liang H, Yuan J, Thalmann D. 3d convolutional neural networks for efficient and robust hand pose estimation from single depth images. InProceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2017 (pp. 1991-2000).

[2] Saxena A, Sun M, Ng AY. Make3D: Depth Perception from a Single Still Image. In AAAI 2008 Jan (Vol. 3, pp. 1571-1576).

[3] Paszke A, Gross S, Chintala S, Chanan G. Pytorch: Tensors and dynamic neural networks in python with strong gpu acceleration. PyTorch: Tensors and dynamic neural networks in Python with strong GPU acceleration. 2017 May;6(3).

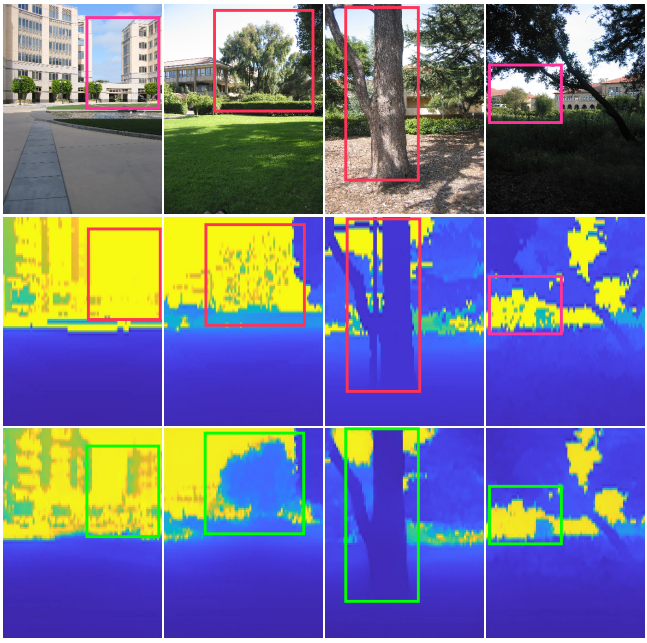

Fig. 11 Qualitative analysis of the proposed model with the Make3D dataset: color images (Row 1), ground-truth depth maps (Row 2), and estimated depth maps (Row 3).

[4] Kingma, Diederik P., and Jimmy Lei Ba. 'ADAM: AMETHOD FOR STOCHASTIC OPTIMIZATION.'

[5] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 'U-net: Convolutional networks for biomedical image segmentation.' International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[6] Achanta, Radhakrishna, et al. 'SLIC superpixels compared to state-of-the-art superpixel methods.' IEEE transactions on pattern analysis and machine intelligence 34.11 (2012): 2274-2282.

[7] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 'Fully convolutional networks for semantic segmentation.' Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[8] Laina, Iro, et al. 'Deeper depth prediction with fully convolutional residual networks.' 2016 Fourth international conference on 3D vision (3DV). IEEE, 2016.

[9] Liu, Fayao, Chunhua Shen, and Guosheng Lin. 'Deep convolutional neural fields for depth estimation from a single image.' Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[10] Eigen, David, Christian Puhrsch, and Rob Fergus. 'Depth map prediction from a single image using a multi-scale deep network.' Advances in neural information processing systems 27 (2014).

[11] Li, Bo, et al. 'Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs.' Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[12] Maas, Andrew L., Awni Y. Hannun, and Andrew Y. Ng. 'Rectifier nonlinearities improve neural network acoustic models.' Proc. icml. Vol. 30. No. 1. 2013.

[13] Choi, Yunjey, et al. 'Stargan: Unified generative adversarial networks for multi-domain image-to-image translation.' Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[14] Alhashim I, Wonka P. High Quality Monocular Depth Estimation via Transfer Learning.

[15] Saxena, Ashutosh, Jamie Schulte, and Andrew Y. Ng. 'Depth Estimation Using Monocular and Stereo Cues.' IJCAI. Vol. 7. 2007.

[16] Saxena, Ashutosh, Sung Chung, and Andrew Ng. 'Learning depth from single monocular images.' Advances in neural information processing systems 18 (2005).

[17] Lin L, Huang G, Chen Y, Zhang L, He B. Efficient and high-quality monocular depth estimation via gated multi-scale network. IEEE Access. 2020 Jan 7;8:7709-18.

[18] Abdulwahab S, Rashwan HA, Cristiano J, Chambon S, Puig D. Effective 2D/3D Registration using Curvilinear Saliency Features and Multi-Class SVM. InVISIGRAPP (5: VISAPP) 2019 (pp. 354-361).

[19] Rashwan HA, Chambon S, Gurdjos P, Morin G, Charvillat V. Using curvilinear features in focus for registering a single image to a 3D object. IEEE Transactions on Image Processing. 2019 Apr 22;28(9):4429-43.

[20] Rashwan HA, Chambon S, Gurdjos P, Morin G, Charvillat V. Towards multi-scale feature detection repeatable over intensity and depth images. In2016 IEEE International Conference on Image Processing (ICIP) 2016 Sep 25 (pp. 36-40). IEEE.

[21] Wang N, Zhang Y, Li Z, Fu Y, Liu W, Jiang YG. Pixel2mesh: Generating 3d mesh models from single rgb images. InProceedings of the European conference on computer vision (ECCV) 2018 (pp. 52-67).

[22] Rashwan, Hatem A., et al. 'Using curvilinear features in focus for registering a single image to a 3D object.' IEEE Transactions on Image Processing 28.9 (2019): 4429-4443.

[23] Xu, Shuzhen, Qing Zhu, and Jin Wang. 'Generative image completion with image-to-image translation.' Neural Computing and Applications 32.11 (2020): 7333-7345.

[24] Silberman N, Hoiem D, Kohli P, Fergus R. Indoor segmentation and support inference from rgbd images. InEuropean conference on computer vision 2012 Oct 7 (pp. 746-760). Springer, Berlin, Heidelberg.

[25] Wiles O, Gkioxari G, Szeliski R, Johnson J. Synsin: End-to-end view synthesis from a single image. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2020 (pp. 7467-7477).

[26] Agarwal N, Chiang CW, Sharma A. A study on computer vision techniques for self-driving cars. InInternational Conference on Frontier Computing 2018 Jul 3 (pp. 629-634). Springer, Singapore.

[27] Andhare P, Rawat S. Pick and place industrial robot controller with computer vision. In2016 International Conference on Computing Communication Control and automation (ICCUBEA) 2016 Aug 12 (pp. 1-4). IEEE.

[28] Kanbara M, Okuma T, Takemura H, Yokoya N. A stereoscopic video see-through augmented reality system based on real-time vision-based registration. InProceedings IEEE Virtual Reality 2000 (Cat. No. 00CB37048) 2000 Mar 18 (pp. 255-262). IEEE.

[29] Pirvu M, Robu V, Licaret V, Costea D, Marcu A, Slusanschi E, Sukthankar R, Leordeanu M. Depth distillation: unsupervised metric depth estimation for UAVs by finding consensus between kinematics, optical flow and deep learning. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021 (pp. 3215-3223).

[30] Schonberger JL, Frahm JM. Structure-from-motion revisited. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 4104-4113).

[31] Lowe, David G. 'Distinctive image features from scale-invariant keypoints.' International journal of computer vision 60.2 (2004): 91-110.

[32] Zheng, Jin, and Lihui Peng. 'An autoencoder-based image reconstruction for electrical capacitance tomography.' IEEE Sensors Journal 18.13 (2018): 5464-5474.

[33] Blendowski, Max, Nassim Bouteldja, and Mattias P. Heinrich. 'Multimodal 3D medical image registration guided by shape encoder–decoder networks.' International journal of computer assisted radiology and surgery 15.2 (2020): 269-276.

[34] Ben Abdallah, Mariem, et al. 'Noise-estimation-based anisotropic diffusion approach for retinal blood vessel segmentation.' Neural Computing and Applications 29.8 (2018): 159-180.

[35] Garg R, Bg VK, Carneiro G, Reid I. Unsupervised cnn for single view depth estimation: Geometry to the rescue. InEuropean conference on computer vision 2016 Oct 8 (pp. 740-756). Springer, Cham.

[36] Abdulwahab, Saddam, et al. 'Adversarial learning for depth and viewpoint estimation from a single image.' IEEE Transactions on Circuits and Systems for Video Technology 30.9 (2020): 2947-2958.

[37] PUIG, Domenec. 'Mgnet: Depth map prediction from a single photograph using a multi-generative network.' Artificial Intelligence Research and Development: Proceedings of the 22nd International Conference of the Catalan Association for Artificial Intelligence. Vol. 319. IOS Press, 2019.

[38] Wofk D, Ma F, Yang TJ, Karaman S, Sze V. Fastdepth: Fast monocular depth estimation on embedded systems. In2019 International Conference on Robotics and Automation (ICRA) 2019 May 20 (pp. 6101-6108). IEEE.

[39] Fu H, Gong M, Wang C, Batmanghelich K, Tao D. Deep ordinal regression network for monocular depth estimation. InProceedings of the IEEE conference on computer vision and pattern recognition 2018 (pp. 2002-2011).

[40] Hao Z, Li Y, You S, Lu F. Detail preserving depth estimation from a single image using attention guided networks. In2018 International Conference on 3D Vision (3DV) 2018 Sep 5 (pp. 304-313). IEEE.

[41] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In2009 IEEE conference on computer vision and pattern recognition 2009 Jun 20 (pp. 248-255). Ieee.

[42] Lehtinen J, Munkberg J, Hasselgren J, Laine S, Karras T, Aittala M, Aila T. Noise2Noise: Learning Image Restoration without Clean Data. InInternational Conference on Machine Learning 2018 Jul 3 (pp. 2965-2974). PMLR

[43] Ramamonjisoa, Michaël, Michael Firman, Jamie Watson, Vincent Lepetit, and Daniyar Turmukhambetov. 'Single Image Depth Estimation using Wavelet Decomposition.' (2021).

[44] Tang, Mengxia, et al. 'Encoder-Decoder Structure With the Feature Pyramid for Depth Estimation From a Single Image.' IEEE Access 9 (2021): 22640-22650.

[45] Karsch, Kevin, Ce Liu, and Sing Bing Kang. 'Depth extraction from video using non-parametric sampling.' European conference on computer vision. Springer, Berlin, Heidelberg, 2012.

[46] Kuznietsov Y, Stuckler J, Leibe B. Semi-supervised deep learning for monocular depth map prediction. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 6647-6655).

[47] Hu, Jie, Li Shen, and Gang Sun. 'Squeeze-and-excitation networks.' Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.

[48] Godard C, Mac Aodha O, Brostow GJ. Unsupervised monocular depth estimation with left-right consistency. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 270-279).

[49] Trelinski, Jacek, and Bogdan Kwolek. 'CNN-based and DTW features for human activity recognition on depth maps.' Neural Computing and Applications 33.21 (2021): 14551-14563.

[50] Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. InProceedings of the IEEE conference on computer vision and pattern recognition 2017 (pp. 1492-1500).

[51] Kostadinov D, Ivanovski Z. Single image depth estimation using local gradient-based features. In2012 19th International Conference on Systems, Signals and Image Processing (IWSSIP) 2012 Apr 11 (pp. 596-599). IEEE.

[52] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 770-778).

[53] Karsch, Kevin, Ce Liu, and Sing Bing Kang. 'Depth transfer: Depth extraction from video using non-parametric sampling.' IEEE transactions on pattern analysis and machine intelligence 36.11 (2014): 2144-2158.

[54] Jun, Jinyoung, et al. 'Monocular Human Depth Estimation Via Pose Estimation.' IEEE Access 9 (2021): 151444-151457.

[55] Ling, Chuanwu, Xiaogang Zhang, and Hua Chen. 'Unsupervised monocular depth estimation using attention and multi-warp reconstruction.' IEEE Transactions on Multimedia (2021).

[56] Luo, Bowen, et al. 'Decomposition algorithm for depth image of human health posture based on brain health.' Neural Computing and Applications 32.10 (2020): 6327-6342.

[57] Wu, Jipeng, et al. 'Fast Monocular Depth Estimation via Side Prediction Aggregation with Continuous Spatial Refinement.' IEEE Transactions on Multimedia (2022).

[58] Ji, Rongrong, et al. 'Semi-supervised adversarial monocular depth estimation.' IEEE transactions on pattern analysis and machine intelligence 42.10 (2019): 2410-2422.

[59] Sun, Hang, et al. 'Scale-free heterogeneous cycleGAN for defogging from a single image for autonomous driving in fog.' Neural Computing and Applications (2021): 1-15.

[60] Shen G, Zhang Y, Li J, Wei M, Wang Q, Chen G, Heng PA. Learning Regularizer for Monocular Depth Estimation with Adversarial Guidance. InProceedings of the 29th ACM International Conference on Multimedia 2021 Oct 17 (pp. 5222-5230).

[61] Liu, Jiaying, et al. 'Retrieval compensated group structured sparsity for image super-resolution.' IEEE Transactions on Multimedia 19.2 (2016): 302-316.

[62] Jain, Deepak Kumar, et al. 'GAN-Poser: an improvised bidirectional GAN model for human motion prediction.' Neural Computing and Applications 32.18 (2020): 14579-14591.

[63] Ding, Yinzhang, et al. 'Digging into the multi-scale structure for a more refined depth map and 3D reconstruction.' Neural Computing and Applications 32.15 (2020): 11217-11228.