Neurocomputing 517 (2023) 81–92

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

GCNDepth: Self-supervised monocular depth estimation based on graph convolutional network

Armin Masoumian [a,b,*], Hatem A. Rashwan [a], Saddam Abdulwahab [a], Julián Cristiano [a], M. Salman Asif [b], Domenec Puig [a]

[a] Department of Computer Engineering and Mathematics, University of Rovira i Virgili, Carretera de Valls, Tarragona 43007, Catalonia, Spain
[b] Department of Electrical and Computer Engineering, University of California, Riverside, 900 University Ave., Riverside 92521, CA, USA

ARTICLE INFO

Article history:
Received 29 March 2022
Revised 4 October 2022
Accepted 15 October 2022
Available online 31 October 2022

Keywords:
Deep learning
Graph convolutional network
Monocular depth estimation
Self-supervision

ABSTRACT

Depth estimation is a challenging task of 3D reconstruction to enhance the accuracy sensing of environment awareness. This work brings a new solution with improvements, which increases the quantitative and qualitative understanding of depth maps compared to existing methods. Recently, convolutional neural networks (CNN) have demonstrated their extraordinary ability to estimate depth maps from monocular videos. However, traditional CNN does not support a topological structure, and they can work only on regular image regions with determined sizes and weights. On the other hand, graph convolutional networks (GCN) can handle the convolution of non-Euclidean data, and they can be applied to irregular image regions within a topological structure. Therefore, to preserve object geometric appearances and objects locations in the scene, in this work, we aim to exploit GCN for a self-supervised monocular depth estimation model. Our model consists of two parallel auto-encoder networks: the first is an auto-encoder that will depend on ResNet-50 and extract the feature from the input image and on multi-scale GCN to estimate the depth map. In turn, the second network will be used to estimate the ego-motion vector (i.e., 3D pose) between two consecutive frames based on ResNet-18. The estimated 3D pose and depth map will be used to construct the target image. A combination of loss functions related to photometric, reprojection, and smoothness is used to cope with bad depth prediction and preserve the discontinuities of the objects. Our method and performance are improved quantitatively and qualitatively. In particular, our method provided comparable and promising results with a high prediction accuracy of 89% on the publicly available KITTI dataset. Our method also offers 40% reduction in the number of trainable parameters compared to the state of the art solutions.In addition, we tested our trained model with Make3D dataset to evaluate the trained model on a new dataset with low resolution images. The source code is publicly available at (https://github.com/ArminMasoumian/GCNDepth.git)

1. Introduction

In the Artificial Intelligence (AI) field, especially deep learning (DL) networks have accomplished high performance in various depth estimation and ego-motion prediction tasks, and nowadays, it is steeply expanding. The importance of depth estimating, as a pull factor for the entry of modern technologies into self-driving vehicles [1,2], object distance prediction [3], helping impaired/blind people, is targeting the improvement of the quality and productivity in the day-to-day life of humankind. The depth estimation based on DL can be utilized in simultaneous localization and mapping (SLAM), navigation, object detection, and semantic segmentation [4].

The stereo vision system is one of the common techniques for depth estimation [5–7]. However, in order to save costs and computational resources, many methods have been presented to perform depth estimation based on a monocular camera. The monocular depth estimation methods can be divided into two categories in terms of the learning approach: supervised learning methods [8,9] and unsupervised learning methods [10,11]. The primitive works focused on studying the extent of the depth prediction with deep supervised networks. Nevertheless, gathering extensive and accurate datasets and ground truth depth for training

* Corresponding author at: Department of Computer Engineering and Mathematics, University of Rovira i Virgili, Carretera de Valls, Tarragona 43007, Catalonia, Spain.
E-mail address: masoumian.armin@gmail.com (A. Masoumian).

https://doi.org/10.1016/j.neucom.2022.10.073
0925-2312/

supervised models is a difficult task [8], especially for developing a ground truth with high resolution and quality. Besides, costly components such as 2D/3D LIDAR sensors are needed to capture depth maps. Thus, many works, such as [12], used time-to-flight cameras to reduce the power for depth sensing to reduce the cost.

Therefore, many depth estimation works were proposed based on unsupervised learning to avoid collecting real depth maps. Most unsupervised learning methods are used to estimate both depth and camera ego-motion [13] to reconstruct the target frame. The main idea is to receive a sequence of frames as an input and to minimize the error between a warped frame and the target one. The warped frame is obtained from an adjacent one, predicted depth, and relative camera motion of the target frame [14]. Most existing DL monocular depth estimation networks use convolutional neural networks (CNN) to extract the feature information and construct the depth images [15,16]. However, CNN is limited, since it does not consider the characteristics of the geometric depth information and object location, as well as contextual features in the scene. Besides, there is recently a need to extend deep neural models from Euclidean domains achieved by CNNs to non-Euclidean domains [17]. Thus, the research community has started to observe the importance of DL networks based on graphs [18]. The effectiveness of the graph convolution network (GCN) has been proved in processing graph data on the tasks of classification [19] and segmentation [20]. For self-supervised monocular depth estimation, there are few works based on GCN, such as depth prediction [21]. The DL model proposed in [21] consists of two stages. The first stage is to use an autoencoder network based on CNNs to estimate the coarse depth and extract latent features of the input image. The second stage is a reconstruction network based on GCNs to refine the predicted depth map. Nonetheless, using two consequent networks leads to an increase in the complexity of the proposed model. Thus, in this work, we propose a novel one stage architectural DL network based on GCN, the so-called GCNDepth, that can help to advance monocular depth estimation.

In general, the two main contributions are summarized as follow:

- We propose a novel autoencoder (CNN-GCN) for monocular depth estimation, which its encoder network is based on ResNet [22] as a backbone to extract key features of the input frame. A decoder network then utilizes the structure of the GCN through the whole decoding process to improve the accuracy of depth maps by learning the nodes (i.e., pixels) representation via constructing the depth maps via iteratively propagating neighbor's information until reaching a stable point.
- To widely exploit the diverse spatial correlations between the pixels at multiple scales and to refine the final estimated depth image by preserving the global information that crosses the coarser feature maps and detailed local information passed from lower layer feature maps, we propose a multi-level depth predictor based on GCN in each layer of the decoder network. The updated graph is fed to the next GCN layer in the decoder network.

In addition, for training the proposed model, we utilize a combination of different loss functions, related to photometric [23], reprojection [14] and smoothness [24] to improve the quality of predicted depth maps. The reprojection loss is used to cope with objects occlusion, and the photometric loss is proposed for feature reconstruction to reduce the losses between target and reconstructed images. In turn, smoothness loss is used to preserve the edges and boundaries of the objects and reduce the effect of texture regions on the estimated depth.

This article is organized as follows, Section two reviews the background and related works on monocular depth estimation, the detailed explanation of the proposed model is described in Section three. The validation of our system through experimental results is given in Section four and Section five represents the conclusion of this research.

2. Background and related work

Monocular depth estimation can be widely categorized into supervised and self-supervised DL. In this section, we present a brief review of both supervised- and self-supervised-based methods. Additionally, we will present depth estimation based on GCN networks.

2.1. Supervised Depth Estimation

Single image depth estimation is an intrinsically ill-posed dilemma: a single input image can project multiple feasible depth maps. Supervised learning methods proved that fitting the relation between color images and their corresponding depth maps by learning ground truth can solve the problem of monocular depth estimation. Diversified approaches have been explored for solving this problem. Fu et al. [25] proposed a multi-layer deconvolution network for obtaining high-resolution depth maps. However, this will require a high computation system for training and create a complicated network. Alhashim et al. [26] proposed a deep learning method via transfer learning to reduce the computation time and network complexity. Their technique contains a standard CNN autoencoder (i.e., encoder-decoder) architecture to estimate high-quality depth maps. However, their depth maps have low pixel resolution, leading to missing depth information in many regions in complex scenes. All fully supervised training approaches require RGB images and the corresponding depth maps as ground truth. However, finding and collecting the original ground truths for supervised training is one of the main limitations. Chen et al. [27] proposed an attention-based context aggregation network (ACAN) to tackle continuous context information capturing problem. Their network improved in detecting the sharp boundaries in the resulting depth maps, but still, they need an original ground truth for labelling and training their model. Real ground truth can be delicately collected from LIDAR sensors or be rendered from simulation engines [9]. However, the LIDAR sensors limit allocating to new vision sensors and rendering real scenes [24]. In general, creating or collecting datasets with accurate depth maps for supervised training models is still challenging. In addition, most supervised-learning models are excellent for specific-purpose environments involved in the datasets, but they can not be easily generalized to different environments. Thus, in this work, we will depend on an unsupervised DL model to estimate the depth maps to cope with the problem of collecting ground-truth depth maps and to discover hidden and interesting patterns in unlabeled images.

2.2. Self-supervised Depth Estimation

As an alternative to the absence of ground truth, self-supervised models can be trained by comparing a target image to a reconstructed image as a supervisory signal. Image reconstruction can be achieved either by stereo training or monocular training.

Stereo training uses synchronized stereo pairs of images and predicts the disparity pixel between the pairs [11]. There are various depth estimation approaches based on stereo pairs. For instance, Garg et al. [10] introduce predicting continuous disparity feature matching framework, which does not require a pre-training stage or annotated ground-truth depths. Their methods consist of a deep CNN autoencoder with inverse warping. However, the gener-

ated view synthesis is only a proxy for depth and may not always yield high-quality predicted depth. In order to improve the constraints of the depth prediction, DL networks need to predict the camera pose between the two consecutive frames (or left–right pairs) during training. Consequently, Madhu et al. [28] used an autoencoder network and a temporal photometric warp error for 6-DoF camera pose estimation (ego-motion) and depth maps. Most of the approaches mentioned above depended on standard loss function, such as L1/L2 norm that yields blurred depth maps. Thus, many trials used robust loss functions to preserve the edges in the predicted depth images. For instance, Zhang et al. [29] proposed a hybrid geometric-refined loss function to explore a more accurate geometric relationship between the input color image and the predicted depth map and preserve depth boundaries and fine structures in depth maps. These approaches rely on geometry to estimate depths from triangulation which often ignores monocular cues (e.g. linear perspective, texture gradients, familiar sizes) [30].

In turn, monocular depth estimation approaches, such as using enforced edge consistency [31], multi-layer feature fusion CNN [32], and adding a depth normalization layer as smoothness term [13], have achieved high performance compared to the stereo pair training. For instance, Wang et al. [33] proposed a multi-task network based on differentiable direct visual odometry, which is fused with an appearance-matching loss to predict depth maps. However, this multi-task model reduces the quality of the predicted depths. Some self-supervised training also makes assumptions about material properties and appearance, such as the brightness constancy of object surfaces between frames [34]. For instance, Liu et al. [35] used domain separation to relieve the illumination variation between day and night images. However, their model needs to learn an illumination-invariant feature space. Most of the models mentioned above tackled the problem self-supervised by learning the depth map based on the photometric error and adopting differentiable interpolation [36–39,14,40] as loss functions. However, these methods often fail to represent the depth boundaries of objects. This problem happens because of an inefficient decoding scheme that causes blurring artifacts at the depth boundary. Consequently, in order to preserve the objects' boundaries and the small details in the predicted depths, new reconstruction strategies are required to build non-Euclidean depth information. It is worth mentioning that networks such as Graph Neural Networks (GNN) can be used to adapt existing monocular depth estimation approaches to directly process and build non-Euclidean structured depth maps.

2.3. Graph Neural Network

Most self-supervised monocular depth estimation methods such as [14,24] depend on CNN-based autoencoder networks to extract visual features from whole scene images and estimate the depth images. However, in most cases, CNN-based networks yield blurred edges and boundaries of the objects. Here, GNN can help capture the dependencies among objects, and GNN can also extract object-based location features from the scene. CNN and GNN models are similar in weight sharing; the main difference is their data structure. Regarding work on data with underlying non-regular structures (irregular on non-Euclidean structured data), the GNN performs better than CNN because the model learns the features by inspecting neighbouring nodes. The basic idea behind most GNN architectures is graph convolution networks (GCN). The GCNs models can learn feature representation even before training because of the adjacency matrix, which helps the model understand adjacent nodes' characteristics [41]. There are two main types of GCNs: Spectral Convolution and Spatial Convolution. The spectral convolution networks use Eigen decomposition of the Laplacian Matrix of the graph. In contrast, spatial convolution networks use the local neighbourhood of nodes and understand the properties of a node based on its k local neighbour, which helps to reduce the computing time significantly without reducing the performance. For the first time, for a semi-supervised node classification task on graphs, GCN was proposed by Kipf et al. [18] for a learning method for the target node to propagate the neighbouring information through convolutional neural networks (ConvGNNs). Their primary purpose of using GCN was to reduce the number of lost features during the feature extraction, help the model learn small details, and predict better results. Their model employed a propagation rule based on graphs' first-order approximation of spectral convolutions. However, this method requires high computational resource consumption depending on the input data size. Another example of a graph-based model based on CNNs was proposed in [42] by arranging the adjacent nodes' information with convolution based on spectral graph theory. However, this causes losing many nodes of the image when 3D objects are mapped in 2D planes.

There are not many works using GCN in monocular depth estimation. Fu et al. [21] created a topological depth graph from a coarse depth map based on spatial graph theory, and they used this graph as a depth clue in their model to avoid depth node losses. However, this technique generates a topological depth graph from a coarse depth map obtained from a pre-trained depth estimation model; thus, their approach consists of two consequent networks. That increase the complexity of the model. In this work, we propose a self-supervised CNN-GCN auto-encoder for monocular depth estimation to solve the above-mentioned problems. The reason for using GCN as a decoder network is to improve the detection of sharp boundaries, reduce the background noise, and compute precise depth maps with full object details compared to the self-supervised state-of-the-art models. Based on [21] and the pixel similarity connection, the graph-based decoder will also sharply detect the edges in depth maps and preserve the small details. In our proposed model, we will use one stage network consisting of an encoder to extract the features of the input image-based CNNs and then a decoder based on GCNs [18] to predict multi-scale depth maps. In addition, our approach will use a combination of different warping errors proposed in the state-of-the-art, such as the reconstruction error presented in [23] to minimize the errors in the reconstructed image, the photometric reprojection error proposed in [14] to optimize the values which provide matching pixel intensities between the target and reconstructed images. Finally, a combination of discriminative and curvature errors [24] to highlight geometric characteristics of the objects and textured regions in the scene. We used the spectral GCN model proposed [18] as a baseline, but we changed it to be as a spatial GCN in order to reduce the computing time without affecting the quality of our results.

3. Method

In this section, we firstly describe the architecture of the proposed model introducing our GCN model and the whole structure of our self-supervised model, including depth estimation (DepthNet) and pose estimation, i.e., ego-motion (PoseNet) networks. Our method will estimate the depth images and the ego-motion to increase the constraints of depth prediction. For monocular depth estimation, the relationship between object location and visual and contextual features in the scene is significant to preserve the objects' boundaries. Besides, we present the loss functions used for training the model.

3.1. Problem Definition

GCNDepth is a multi-task DL-based system that consists of two parallel networks, DepthNet and PoseNet. If $I \in \mathbb{A}$ represent a

monocular RGB image, the problem of generating its corresponding depth image, $D \in \mathbb{B}$, can be formally defined as a function $\Psi_D : \mathbb{A} \longrightarrow \mathbb{B}$ that maps elements from domain $\mathbb{A}$ to elements in its corresponding domain $\mathbb{B}$, as follows:

$$D = \Psi_D(I_s), \tag{1}$$

where the proposed model, DepthNet, approximates the prediction of a depth map, D, as a function, Ψ_D, which is fed by a source RGB frame, I_s as an input with pixels p.

Similarly, the problem of estimating the viewpoint between two consequent RGB images can be formally define as a function $\Psi_E : \mathbb{A} \longrightarrow \mathbb{R}^3$, which is fed by two consequent frames, I_s and I_s as an input and predicts an ego-motion vector, as follows: $E_{I_s \longrightarrow I_t} = [r^T, t^T]$, where $r = [\Delta\theta, \Delta\phi, \Delta\psi]^T$is is a rotation vector, and $t = [\Delta x, \Delta y, \Delta z]^T$is a translation vector. The mapping process can be approximated as follows:

$$E_{I_s \longrightarrow I_t} = \Psi_E(I_s, I_t). \tag{2}$$

Both the depth and ego-motion vector along with the I_s source frame are used for reconstructing an image, I_{rec} that has to be close to the target image, I_t. Thus, our model, GCNDepth, aims at approximating the total process for estimating depth and pose with the I_{rec} in a final function Ψ that accepts two inputs I_s and I_t, as follows:

$$\Psi(I_s, I_t) = (D, E_{I_s \longrightarrow I_t}, I_{rec}). \tag{3}$$

3.2. Graph Convolutional Network

One of the main problems with CNN-based networks is that they cannot compute the data of non-Euclidean domains, and they extract appearance features rather than object-location features. The use of DL models based on CNN on complex 3D scenes, such as depth maps estimation, can yield a significant loss in the details of the objects in the scene or even break the topological structure of the scene [17]. Thus, GCN networks that introduce topological structure and node features can increase the feature representation of hidden layers. That helps the model learn how to map the depth information from low-dimensional features. Besides, they can represent the topological structure of the scene by describing the relations between objects allowed. Generally, the graph convolution is defined as follows:

$$Z = \sigma(AXW), \tag{4}$$

where $\sigma(.)$ defines a non-linear activation function, $\mathbf{A} \in R^{N \times N}$ is an adjacency matrix (i.e., binary matrix), with N being the number of nodes in the given graph that measures the relationship between the nodes in the graph. $\mathbf{X} \in R^{N \times C}$ represents the input N nodes into the graph, and the feature vector dimensionality C, which in our case is high-level (latent) features extracted from the CNN-based encoder. In turn, $\mathbf{W} \in R^{H \times F}$ is the trainable weights, where H is the number of the nodes in the hidden layer and F is the dimensions of the resulting vector. Note that $H = C$ in the first layer.

To avoid the adjacency matrix changing the scale of the feature vector, we added an identity matrix I to obtain the self-loop as follows:

$$\widehat{A} = A + I. \tag{5}$$

In our case and regarding the non-linear activation function, for the first layer of GCN, the ReLU activation is used to reduce the dependency of the parameters and avoid over-fitting. For the second layer of GCN, Log-Softmax is used to normalize the output of the graph. Fig. 1 illustrates the architecture of our GCN model. We randomly initialized the first adjacency matrix of the first graph in the decoder network with the exact size of the nodes in the first layer of the depth decoder. We fine-tuned a parameter, P, which represents the probability for edge creation and the percentage similarity of each node (i.e., vertices) or pixel with their neighbor nodes in the graph. Using $P = 0.7$ with the first random adjacency matrix yields the best estimated depth maps.

In the end, in order to boost and increase the quality of predicted depth maps, a multi-scale GCN based is used in the decoder network. This technique combines the feature information of each scale with a depth graph topology.

3.3. Self-supervised CNN-GCN Autoencoder

To predict the depth map of a single image, the self-supervised training depth estimation network of our model, DepthNet, is an autoencoder network. The autoencoder network consists of two successive sub-networks: the first one is an encoder that maps the input into high-level feature representation and a decoder that maps the feature representation to a reconstruction of the depth. In this work, we proposed to use a CNNs-based encoder and a GCN-based decoder.

3.3.1. DepthNet Encoder

For the encoder network, the input is an image represented as grid-like data, which is regular, and its pixels have the same amount of neighbors. CNNs can exploit the local connectivity and global structure of image data by extracting meaningful local features shared within the input images used during the training stage. Therefore, in our case, CNNs are suitable for extracting global-based visual features from the whole scene shown in the input image. Our encoder network consists of 5 deep layers. The

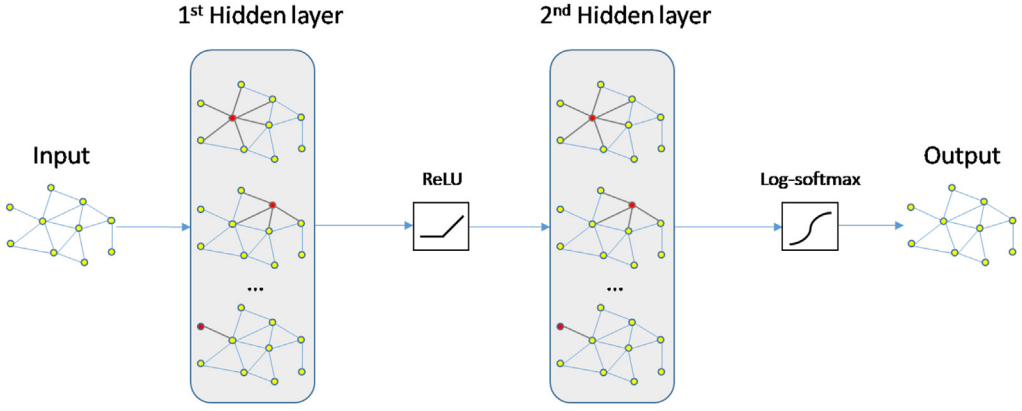

Fig. 1. An illustration of the proposed GCN module containing two hidden layers.

last four layers are standard ResNet-50 [22] blocks. The first layer before the ResNet blocks is a fast convolutional layer, Conv1x1, which consists of a convolution + batch normalization + max-pooling operation. Table 1 represents the network details of the encoder network.

3.3.2. DepthNet Decoder

Regarding the depth decoder and for large-scale depth estimation, we aim to use a geometric DL network that can help extract object-based location features and keep the relationships between nodes in the resulting depth maps by generating a topological depth graph in multi-scale. Therefore, we used multi-scale GCN as shown in Fig. 2. The adjacency matrix of the initial graph is built based on the number of nodes of the features generated by the last layer of the encoder network.

Our approach is to use four levels of GCN in constructing the depth images. The main components of the decoder network are 'upconvolution' layers, consisting of unpooling (up-sampling the feature maps, as opposed to pooling) and a transpose convolution that performs an inverse convolution operation. To accurately estimate the depth images, we apply the 'upconvolution' to feature maps and concatenate it with corresponding feature maps from the corresponding layers of the encoder network and an up-sampled coarser depth prediction using GCN of the previous layer. This approach helps the proposed model preserve the high-level information passed from coarser feature maps and the fine local information provided in lower-layer feature maps. Each step increases the resolution twice. This process is repeated four times, providing a predicted depth map, which is half of the input image. This loop cycle is called multi-scale because, in each layer of our decoder network, the GCN is updated and up-sampled, and is sent to the next layer. The parameters of each layer used in our depth decoder are described in Table 2.

3.3.3. PoseNet Estimator

The pose estimation network is a regression network with encoder and decoder parts. The pose encoder receives a concatenated pair of images, I_s and I_t. Our encoder network consists of 5 deep layers; the first layer is a fast convolutional layer consisting of a 1×1 convolution fed by a concatenation of a pair of images, I_s and I_p, followed by batch normalization and max-pooling. The last four layers are standard ResNet-18 blocks [22], which is similar to our depth encoder with fewer hidden layers. The output of the last layer (i.e., ResNet-18-L4) from the pose encoder is a 512 feature map. In turn, our pose decoder contains four convolution layers. The input of the pose decoder is the output of ResNet-18-L4. Besides, the pose decoder has a convolutional weight in the first layer similar to that proposed in [14]. The decoder layer parameters are shown in Table 3.

3.4. Overall Pipelines

The proposed method consists of two main networks. The first network, called DepthNet explained in the previous subsection. The source image is an input of the DepthNet, and the output is the depth map. The second network is PoseNet, a pose predictor to estimate the ego-motion vector of the source and the target images (in our case, a consecutive image). The output of PoseNet is the relative pose between the source and target images. Afterward, a warping process, as proposed in [14], is applied for finding the corresponding pixels in the adjacent frames through the estimated depth map of the source frame and the camera ego-motion vector, and then synthesize the target frame. These two main networks provide geometry information to provide point-to-point correspondences of the reconstructed image. The whole architecture of our model is illustrated in Fig. 3.

3.5. Geometry Models and Losses

In monocular video datasets, based on the source frame I_s and the target frame I_t, the reconstructed image I_{rec} can be reconstructed using the resulting depth and the 3D pose. The total loss for the whole network contains three main losses, which penalizes the losses between reconstructed and target images on one side and the resulting depth and the source image on the other side.

The first loss function called the reconstruction loss L_{Rec}, is a common context loss function for an autoencoder network used for constraining the quality of the learned features to reconstruct the target image, as proposed in [14]. Thus we used the mean square error between the source and the reconstructed images, as:

Table 1
The network architecture of depth encoder. **K** is the number of block repetition, **S** the stride, **Chn** the number of output channel, **Input** corresponds to the input channel of each layer.

Layer	K	S	Chn	Input	Activation
Conv1x1	1	1	64	Img (1024×320×3)	ReLU
ResNet-50 L1	3	1	256	Conv1x1 (512×160×64)	ReLU
ResNet-50 L2	4	1	512	ResNet L1 (256×80×256)	-
ResNet-50 L3	6	1	1024	ResNet L2 (128×40×512)	-
ResNet-50 L4	3	1	2048	ResNet L3 (64×20×1024	SoftMax

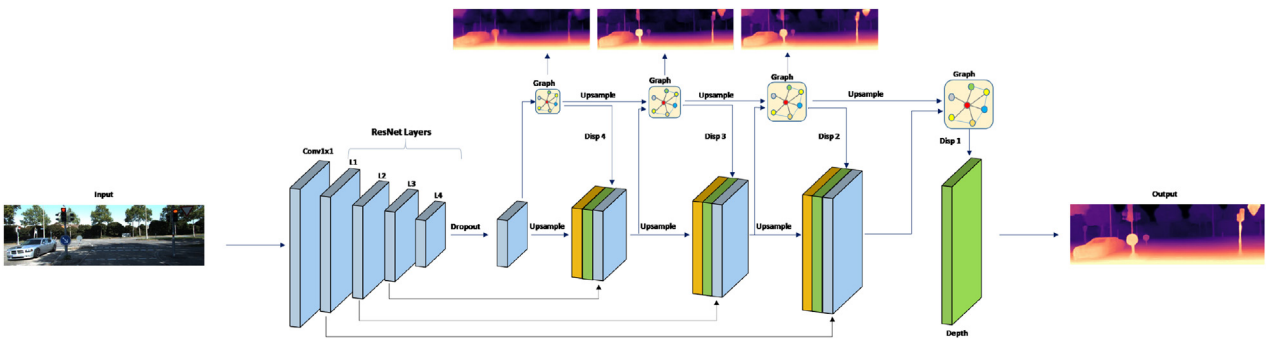

Fig. 2. Overview of DepthNet network architecture.

Table 2
The network architecture of depth decoder. **K** is the kernel size, **S** the stride, **Chn** the number of output channel, **Input** corresponds to the input channel of each layer and ↑ represents upsampling by 2x.

Layer	K	S	Chn	Input	Activation
iL4	3	1	512	L4	Leaky-ReLU
GC4-1	3	1	1	Adj4, iL4	ReLU
GC4-2	3	1	1	Adj4, GC4-1	Log-SoftMax
Disp4	3	1	1	GC4-2	Sigmoid
iL3	3	1	256	L3	Leaky-ReLU
Adj3	3	1	1	Adj4	-
Disp4	3	1	1	Disp4	-
GC3-1	3	1	1	Adj3, iL3, Disp4	ReLU
GC3-2	3	1	1	Adj3, GC3-1	Log-SoftMax
Disp3	3	1	1	GC3-2	Sigmoid
iL2	3	1	128	L2	Leaky-ReLU
Adj2	3	1	1	Adj3	-
Disp3	3	1	1	Disp3	-
GC2-1	3	1	1	Adj2, iL2, Disp3	ReLU
GC2-2	3	1	1	Adj2, GC2-1	Log-SoftMax
Disp2	3	1	1	GC2-2	Sigmoid
iL1	3	1	64	L1	Leaky-ReLU
Adj1	3	1	1	Adj2	-
Disp2	3	1	1	Disp2	-
GC1-1	3	1	1	Adj1, iL1, Disp2	ReLU
GC1-2	3	1	1	Adj1, GC1-1	Log-SoftMax
Disp1	3	1	1	GC1-2	Sigmoid

Table 3
The network architecture of pose decoder. **K** is the kernel size, **S** the stride, **Chn** the number of output channel and **Input** corresponds to the input channel of each layer.

Layer	K	S	Chn	Input	Activation
Out1	1	1	256	ResNet-18 L4	ReLU
Out2	3	1	256	Out1	ReLU
Out3	3	1	256	Out2	ReLU
Out4	1	1	6	Out3	-

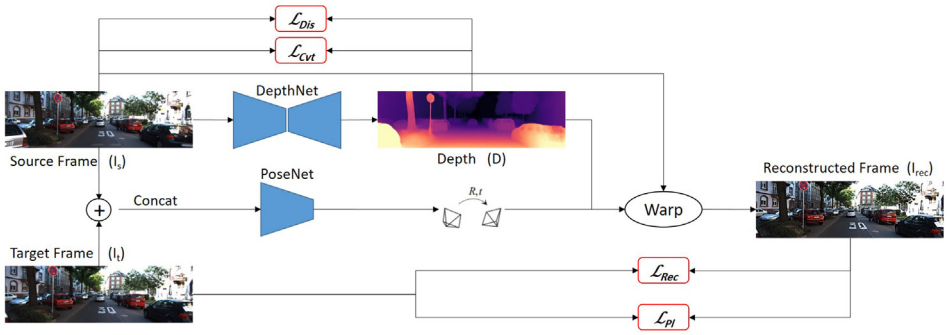

Fig. 3. Schematic illustration of the whole framework.

$$L_{Rec} = \sum_p \sqrt{(I_{rec}(p) - I_t(p))^2}. \tag{6}$$

Regarding achieving better performance and coping with occlusions between frames in a monocular video, the reconstruction loss L_{Rec} is combined with the reprojection loss, L_{PI}, which combines the L1-norm and SSIM losses as defined in [13].

$$L_{PI} = 0.15\sum_p |I_{rec}(p) - I_t(p)| + 0.85\sum_p \frac{1 - SSIM(I_{rec}, I_t)}{2} \tag{7}$$

In addition, if we consider that the image intensity function obeys the Lambertian shading function, the network should extract gradient-based features corresponding to the object's shapes in the input color image. To handle the depth discontinuity is usually problematic due to occlusion, over-smoothing, and textured regions, the resulting depth map requires a loss function to preserve the edges and boundaries of the objects and degrade the texture effects. Thus, the first and the second derivative of depth images can highlight geometric characteristics of the objects and homogeneous regions in the image [43]. Consequently, to ensure that the learned features of the input image yield edge-preserving depth maps, a discriminative loss function, L_{Dis}, can be defined to give significant weight to the low texture regions.

$$L_{Dis} = \sum_p e^{-\lambda \nabla^1 I_s(p)} |\nabla^1 D(p)|, \tag{8}$$

where, D represents the predicted depth maps at each pixel p, ∇^1 represents the first order derivative at each pixel p and λ, a weight factor, is empirically set by 0.5 in this work that yielded the highest accuracy.

In addition, the second-order behavior of the surface in a scene is compatible with the curvature measurements of the depth sur-

face relative to the normal at one of its points near this point. Thus, a curvature loss L_{Cvt} can be defined based on the second-order derivative of gradients as proposed in [24]. L_{Cvt} also keeps the geometric characteristics of the objects and gives a low weight for textured regions:

$$L_{Cvt} = \sum_{p} e^{-\lambda \nabla^2 I_s(p)} |\nabla^2 D(p)|. \quad (9)$$

The combination of discriminative and curvature losses is used as a smoothness loss function which can be defined as:

$$L_{Smooth} = \alpha L_{Dis} + \beta L_{Cvt}. \quad (10)$$

The α and β are set to $1e-3$ via cross validation as proposed in [24].

The final loss can be used for the optimization process of the whole network, and a penalty for a lousy depth prediction is defined as:

$$L_{Final} = L_{PI} + L_{Rec} + L_{Smooth}. \quad (11)$$

3.6. Implementation Details

We implemented our method by using the PyTorch framework [44], and the proposed model was trained for 20 epochs with a batch size of 10 with one GTX 1080-TI GPU. The Adam optimizer [45] has been utilized with an initial learning rate of 0.0001 and reduced by half after 75% of the total iterations. The pre-trained ResNet-18 and ResNet-50 layers are used for the PoseNet and DepthNet encoders, respectively [46].

4. Experiments

In this section, we demonstrate the evaluation performance of our proposed model. To evaluate our approach, we carry out comprehensive experiments on public benchmark datasets such as KITTI dataset [47] and Make3D dataset [48].

4.1. Depth Evaluation on the KITTI Dataset

KITTI dataset is a vision dataset for depth and poses estimation. The dataset contains 200 videos of street scenes in the daylight captured by RGB cameras and the depth maps captured by the Velodyne laser scanner. The synchronized single images from a monocular camera were used and Eigen split [49] with 39810 images for training, 4424 for validation, and 697 images for testing. The image pre-processing method proposed in [36] has been used for removing static frames. The resolution of the images is 1024×320 pixels.

Regarding the evaluation, we used the standard metrics of depth evaluation, such as Absolute and Relative Error (Abs-Rel), Squared Relative Error (Sq-Rel), Root Mean Squared Error (RMSE), and Root Mean Squared Log Error (RMSE-Log). Besides, we used δt to calculate the accuracy of the estimated depth with different thresholds as proposed in [47].

The same original input image size is used for evaluation and depth is capped at 80 meters based on the information from the KITTI dataset. Both input size and output size of images are 1024×320 pixels.

The median scaling introduced by [24] is used for predicted depths to match the ground-truth scale. The median scaling is multiplying predicted depth maps by a computed scale factor to match the median with the ground truth. A different scaling factor is calculated for each test image individually. The proposed framework is compared with the state-of-the-art of self-supervision based monocular depth estimation [14,24,31,36,50,39,37,51,38,52–55,40,56]. Where [36,50,39,53] used DispNet [9] as a backbone for the encoder network. DispNet is a network used a standard CNN to build the encoder and decoder network to find the disparity between two successive or stereo images. In turn [31,40,14] exploited ResNet-18 and [38,52] used VGG as a backbone. In addition, ResNet-50 was used in [24,37,51,54,57,58] and GCNDepth (our proposed model). The performances of our model compared with the state-of-the-art solutions is summarized in Table 4. The all tested models shown in Table 4 are trained with unsupervised learning and monocular images with a resolution of 1024×320. As shown in Table 4, the GCNDepth method achieved the highest performance in terms of Abs-Rel, Sq-Rel, second and third accuracy of (δ_2,δ_3) evaluation metrics. In addition, the proposed method also achieved second-best results in RMSE, RMSE-Log, and first accuracy of (δ_1) with a slight difference of 0.003 with RMSE-log, and 0.5% with δ_1 compared to the highest results achieved by [24]. In general, the model of Featdepth [24] and our model, GCNDepth, provided comparable results and they outperform the other tested methods.

Although the Featdepth model achieved similar results to our model, the GCNDepth model yields a 40% reduction in the number of trainable parameters compared to the Featdepth model. Where the GCNDepth model has trainable parameters of $48,220,954$, in turn the Featdepth model has $79,681,406$. Since the Featdepth

Table 4
Comparison of different methods on KITTI dataset. Best results are in 'bold' and second best results are in 'italic'.

Method	Backbone	Lower Better				Higher Better		
		Abs-Rel	Sq-Rel	RMSE	RMSE-Log	δ_1	δ_2	δ_3
SfMLearner[36]	DispNet	0.208	1.768	6.958	0.283	0.678	0.885	0.957
DNC[50]	DispNet	0.182	1.481	6.501	0.283	0.725	0.906	0.963
Vid2Depth[39]	DispNet	0.163	1.240	6.220	0.250	0.762	0.916	0.968
LEGO[31]	ResNet-18	0.162	1.352	6.276	0.252	0.783	0.921	0.969
GeoNet[37]	ResNet-50	0.155	1.296	5.857	0.233	0.793	0.931	0.973
DF-Net[51]	ResNet-50	0.150	1.124	5.507	0.223	0.806	0.933	0.973
DDVO[38]	VGG	0.151	1.257	5.583	0.228	0.810	0.936	0.974
EPC++[52]	VGG	0.141	1.029	5.350	0.228	0.816	0.941	0.976
Struct2Depth[53]	DispNet	0.141	1.036	5.291	0.215	0.816	0.945	0.979
SIGNet[54]	ResNet-50	0.133	0.905	5.181	0.208	0.825	0.947	0.981
CC[55]	DispNet	0.140	1.070	5.326	0.217	0.826	0.941	0.975
LearnK[40]	ResNet-18	0.128	0.959	5.232	0.212	0.845	0.947	0.976
PackNet[57]	ResNet-50	*0.107*	0.802	4.538	0.186	*0.889*	*0.962*	0.981
DualNet[56]	HRNet	0.121	0.837	4.945	0.197	0.853	0.955	*0.982*
SimVODIS[58]	ResNet-50	0.123	0.797	4.727	0.193	0.854	0.960	**0.984**
Monodepth2[14]	ResNet-18	0.115	0.882	4.701	0.190	0.879	0.961	*0.982*
FeatDepth[24]	ResNet-50	**0.104**	*0.729*	**4.481**	**0.179**	**0.893**	**0.965**	**0.984**
GCNDepth	ResNet-50	**0.104**	**0.720**	*4.494*	*0.181*	*0.888*	**0.965**	**0.984**

model has an extra deep feature network for feature representation learning to cope with the geometry problem of self-supervision depth estimation. The comparable results show that the use of GCN in reconstructing the depth images can improve the photometric error that appeared in the self-supervision problem without using the feature network as proposed in [24].

In addition, our model achieved high performance on the KITTI benchmark evaluation in the SILog and iRMSE metrics and achieved comparable results in the Sq-Rel and Abs-Rel metrics compared to other state-of-the-art of self-supervised methods as shown in Table 5. The results have shown in Table 5 supported that the use of GCN in estimating depth maps from a monocular video can yield depth maps outperforming or matching the state of the art on the KITTI dataset.

Qualitatively, the comparison of predicted depth results of the proposed model can be seen in Fig. 4. The first row of Fig. 4 represents a clear depth estimation of far and small objects with our GCNDepth model compared to the two methods [24,14]. In the second row of Fig. 4, our method estimates the depth between the consecutive cars and correctly detects the boundaries of the two cars. In the third and fourth row, our method properly preserves the discontinuities of the objects without any distortion which occurred with the two other methods. In the last row of Fig. 4, our model is be able to detect the human body in its full shape showing the depth of the key points of body parts, such as the head, neck, shoulder, etc. However, the other models proposed in [24,14], could not be able to detect the head of the human and there are no homogeneous depth values for other body parts. The qualitative results support that GCNDepth can extract precise depth maps and recover the depth of objects with higher precision compared to the baselines [24,14]. The depth maps generated by GCNDepth maintain the boundaries and details of objects that can be clearly realized. In contrast, depth maps resulting from baselines have crumbled boundaries and the objects can not be recognized. The preservation of the objects' discontinuities can help in building more accurate semantic maps and visual-inertial odometry for autonomous vehicles.

4.2. Ablation Study

To get a better understanding of the performance of the proposed method, in 6, we showed an ablation study by changing different components of the proposed model, GCNDepth, as follows:

- Baseline, which is similar to our model with a CNN-based decoder instead of GCN with different losses.
- Single-scale GCN, (SS), where a single scale was added on the first layer of depth decoder.

Table 5
Performance of our model on KITTI public benchmark.

Method	SILog	Sq-Rel	Abs-Rel	iRMSE
GCNDepth	**15.54**	4.26	12.75	**15.99**
packnSFMHR[57]	15.80	4.75	**12.28**	17.96
MultiDepth[59]	16.05	**3.89**	13.82	18.21
LSIM[60]	17.92	6.88	14.04	17.62

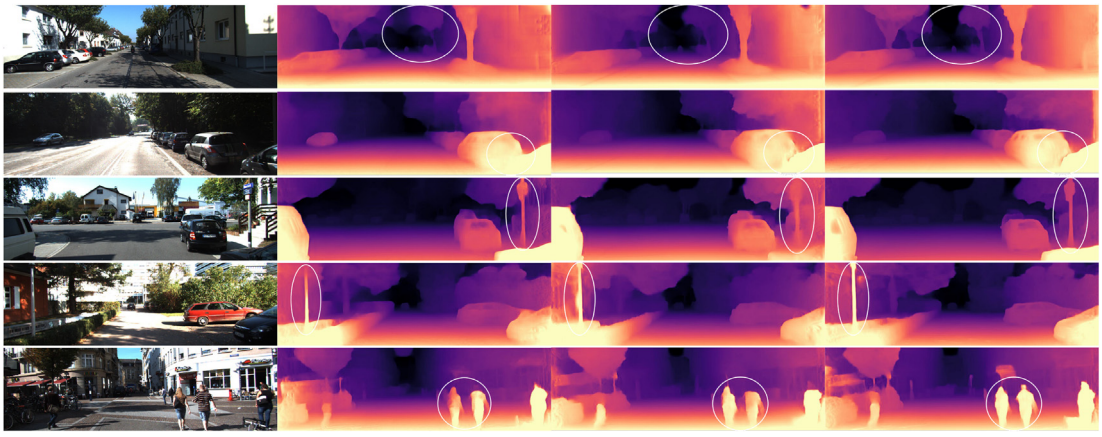

Fig. 4. Comparison of disparity results on KITTI dataset. (Col.1) original input images, and the depth resulted with (Col.2) Monodepth2 [14], (Col.3) FeatDepth [24] and (Col.4) the proposed. GCNDepth model.

Table 6
Ablation results for different components. **SS** represents the single scale GCN and **MS** represent the multi scale GCN.

Methods and Losses	Asb-Rel	Sq-Rel	RMSE	RMSE-Log	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$
Baseline-Res18 (L_{Rec})	0.132	1.052	5.649	0.237	0.791	0.922	0.959
Baseline-Res18 (L_{Rec}+L_{Pl})	0.119	0.880	4.689	0.204	0.877	0.961	0.981
Baseline-Res18 (L_{Rec}+L_{Pl}+L_{Smooth})	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Ours-MS-Res50 (L_{Rec})	0.111	0.867	5.109	0.198	0.853	0.959	0.980
Ours-MS-Res50 (L_{Rec}+L_{Pl})	0.107	0.748	4.635	0.199	0.881	0.960	0.981
Ours-SS-Res50 (L_{Rec}+L_{Pl}+L_{Smooth})	0.135	0.991	5.148	0.213	0.814	0.939	0.977
Ours-MS-Res18 (L_{Rec}+L_{Pl}+L_{Smooth})	0.105	0.739	4.585	0.191	0.883	0.961	0.982
Ours-MS-Res50 (L_{Rec}+L_{Pl}+L_{Smooth})	**0.104**	**0.720**	**4.494**	**0.181**	**0.888**	**0.965**	**0.984**

Table 7
Maked3D results. Type **D** represents depth supervision methods and type **M** represents self-supervised mono supervision.

Method	Type	Abs_Rel	Sq_Rel	RMSE	$\log_{10}$
Karsch[62]	D	0.428	5.079	8.389	0.149
Liu[63]	D	0.475	6.562	10.05	0.165
Laina[61]	D	**0.204**	**1.840**	**5.683**	**0.084**
Zhou[36]	M	0.383	5.321	10.47	0.478
DDVO[38]	M	0.387	4.720	8.090	0.204
Monodepth2[14]	M	**0.322**	3.589	7.417	0.201
GCNDepth	M	0.424	**3.075**	**6.757**	**0.107**

- Multi-scale GCN layers (MS) with different losses.
- GCN network with different pre-trained backbones (i.e., ResNet-18 and ResNet-50).

As shown in 6, we tested our baseline model with three different loss combinations (reconstruction loss (L_{Rec}), photometric loss (L_{PI}), and smoothness loss (L_{Smooth})), GCN model with three different loss combinations, single scale GCN and multi-scale GCN, and different pre-trained backbones of ResNet-18 and ResNet-50. Adding the photometric loss leads to improving the Asb-Rel with 0.13 and it yields a significant improvement of 8% in the accuracy δ comparing to the baseline. Furthermore, adding smoothness loss, besides improving the quality of visual depths, improves the Asb-Rel by 0.04. The single-scale GCN results were not good compared to the baseline, however, the multi-scale GCN with the three loss functions and using a pre-trained model of ResNet-50 achieved higher results compared to the other variations of the proposed models. Using the ResNet-50 instead of ResNet-18, improved the results and accuracy slightly.

Regarding the structure of GCN, we changed the activation function of the GCN layer after the second hidden layer by ReLU or Log-softmax. Besides, we changed the P value for the initialization of the random graph. The experiments showed that multi-scale GCN with a P value of 0.7 (70 percent of similarity) achieved accurate quantitative results than using other values of P, such as $P = 0.1, 0.3, 0.5$ and 0.9. Regarding the activation functions, the proposed final model with multi-scale GCN has achieved the highest score with Log-softmax as an activation function.

4.3. Depth Evaluation on the Make3D Dataset

Additionally, we tested the performance of the GCNDepth model on the Make3D dataset using our trained model based on the KITTI dataset. In other words, we used the Make3D dataset

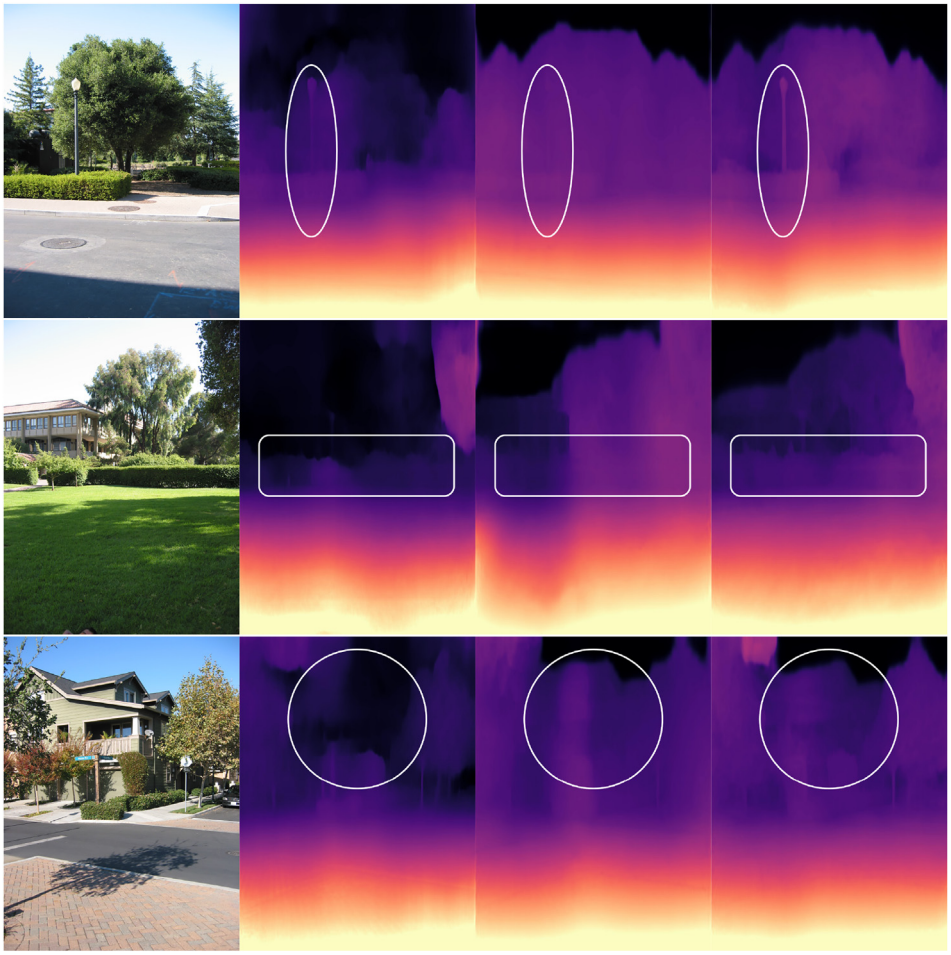

Fig. 5. Comparison of disparity results on Make3D dataset. (Col.1) original input images, and the depth resulted with (Col.2) Monodepth2 [14], (Col.3) FeatDepth [24] and (Col.4) the proposed. GCNDepth model.

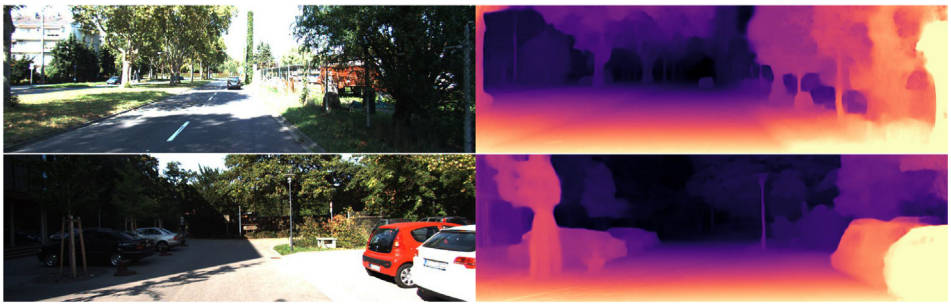

Fig. 6. Two examples of low quality predicted depths.

purely for validation and testing. The Make3D dataset contains 400 RGB images for training and 134 images for a test set. The results in Table 7 show that we outperformed the state-of-the-art of self-supervised methods [14,38,36] evaluated on the Make3D dataset in terms of Sq-Rel, RMSE, and RMSE-log metrics of $3.075, 6.757$ and 0.107, respectively without fine-tuning the GCNDepth model with the training set of Make3D. In turn, the Monodepth2 model [14] yielded the best Abs-Rel error among the four self-supervised approaches with a value of 0.322. While GCNDepth provided the second best Abs-Rel error of 0.424. Besides, the GCNDepth model yielded the second-best results after the supervised-based model proposed in [61], which provided the best results with differences of $0.22, 1.235, 1.075$ and 0.023 of the four metrics: Abs-Rel, Sq-Rel, RMSE, and RMSE-log, respectively. These can be considered promising results compared to the supervised-based approaches.

Qualitative results with the Make3D dataset are shown in Fig. 5. GCNDepth is able to estimate depth values even in low texture regions and with different illumination, changes compared to the two other self-supervision models [24,14]. For instance, in the first row of Fig. 5, compared to the two other models, the depth map resulting from our model showed that the column of the light in the input image is more visible and with homogeneous depth values and closer to the camera than the other objects (e.g., trees). In turn, the second row of Fig. 5 shows that the green view in the image is faded into the background in the depth maps from the baselines, but with our model, the green view in the depth image can be clearly recognized and with boundaries distinguished from the background. In contrast to the other methods in the last row of Fig. 5, the house can be easily identified in the depth map resulting with GCNDepth. In the graph network, the relationships between nodes are of importance that constitute the path of information transmission in GCN. Thus, we believe that the features extracted from GCNs maintain the weights of different objects in the scenes and these features help deal with reconstructing depth maps preserving the discontinuities of the objects. It is obvious that this can possibly improve the performance of reconstructing geometric information for more accurate depth map prediction.

4.4. Limitations

Despite achieving comparable and promising results with the GCNDepth model, the model still has some limitations. Firstly, GCN is inefficient in updating the nodes' hidden states iteratively for a fixed number of the feature vector dimension. However, we can get a stable representation of the node and its neighbourhood by designing a multi-layer GCN as we proposed in this work. Secondly, we must create a random graph with a connection edge probability between each pixel and neighbours for the initial graph. The randomization may increase the training time. Lastly, increasing the number of layers in GCN increases the training time and complexity of the model. In addition, Fig. 6 shows that the image's shadow badly affects the estimated depth that cannot show the small details of the objects and accurate boundaries between the objects. The reason is that the shadows cause the region to become textureless in the image, and the similarities between pixels are high. We aims to develop a model that can cope with the illumination distribution in the images.

5. Conclusion

This paper presents a self-supervised DL model for monocular depth estimation based on a multi-scale graph convolutional network (GCN). The proposed model consists of two networks: 1) depth estimation and 2) pose estimation. The use of GCN in the decoder of the depth estimation auto-encoder can map the depth information from low-dimensional features. It can represent the topological structure of the scene by describing the relations between the scene pixels. Besides, to improve the depth estimation, a combination of different loss functions is used i) absolute mean error between the target image and the reconstruction image, ii) perceptional loss to minimize the photometric reprojection error, and iii) a combination between discriminative and curvature losses to highlight geometric characteristics of the objects and textured regions in the image. The proposed method achieved a comparable depth estimation from monocular video single image to the existing KITTI and Make3D datasets. The generated depth maps with GCNDepth depict object edges and boundaries, helpful for semantic maps and visual odometry. The ongoing work is to improve the network that can predict depth maps for night-time images. In turn, future work aims at developing a complete model for pose, depth, and motion estimation from monocular videos.

CRediT authorship contribution statement

Armin Masoumian: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing - original draft. **Hatem A. Rashwan:** Validation, Data curation, Writing - review & editing. **Saddam Abdulwahab:** Visualization, Investigation. **Julián Cristiano:** Supervision, Formal analysis, Methodology. **M. Salman Asif:** Software, Validation, Supervision, Writing - review & editing. **Domenec Puig:** Resources, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research has been possible with the support of the Secretariad Universitatsi Recercadel Departamentd Empresai Coneixement de la Generalitat de Catalunya (2020 FISDU 00405). We are thankfully acknowledging the use of the University of Rovira I Virgili (URV) facility in carrying out this work. Thanks to Alexis Singh for proofreading the article.

References

[1] C. Badue, R. Guidolini, R.V. Carneiro, P. Azevedo, V.B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T.M. Paixao, F. Mutz, et al., Self-driving cars: A survey, Expert Systems with Applications 165 (2021) .
[2] M. Daily, S. Medasani, R. Behringer, M. Trivedi, Self-driving cars, Computer 50 (12) (2017) 18–23.
[3] A. Masoumian, D. Marei, S. Abdulwahab, J. Cristiano, D. Puig, H.A. Rashwan, Absolute distance prediction based on deep learning object detection and monocular depth estimation models, Artificial Intelligence Research and Development: Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence, Vol. 339, IOS Press, 2021, p. 325.
[4] C. Zhao, Q. Sun, C. Zhang, Y. Tang, F. Qian, Monocular depth estimation based on deep learning: An overview, Science China Technological Sciences 63 (9) (2020) 1612–1627.
[5] H. Laga, L.V. Jospin, F. Boussaid, M. Bennamoun, A survey on deep learning techniques for stereo-based depth estimation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020).
[6] Y. Ming, X. Meng, C. Fan, H. Yu, Deep learning for monocular depth estimation: A review, Neurocomputing 438 (2021) 14–33.
[7] W. Gan, P.K. Wong, G. Yu, R. Zhao, C.M. Vong, Light-weight network for real-time adaptive stereo depth estimation, Neurocomputing 441 (2021) 118–127.
[8] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, Advances in neural information processing systems 27 (2014).
[9] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox, A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 4040–4048.
[10] R. Garg, V.K. Bg, G. Carneiro, I. Reid, Unsupervised cnn for single view depth estimation: Geometry to the rescue, in: European conference on computer vision, Springer, 2016, pp. 740–756.
[11] J. Xie, R. Girshick, A. Farhadi, Deep3d: Fully automatic 2d-to-3d video conversion with deep convolutional neural networks, in: European conference on computer vision, Springer, 2016, pp. 842–857.
[12] J. Noraky, V. Sze, Low power depth estimation of rigid objects for time-of-flight imaging, IEEE Transactions on Circuits and Systems for Video Technology 30 (6) (2019) 1524–1534.
[13] C. Godard, O. Mac Aodha, G.J. Brostow, Unsupervised monocular depth estimation with left-right consistency, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 270–279.
[14] C. Godard, O. Mac Aodha, M. Firman, G.J. Brostow, Digging into self-supervised monocular depth estimation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 3828–3838.
[15] S. Abdulwahab, H.A. Rashwan, M.Á. García, M. Jabreel, S. Chambon, D. Puig, 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.
[16] S. Abdulwahab, H.A. Rashwan, A. Masoumian, N. Sharaf, D. Puig, Promising depth map prediction method from a single image based on conditional generative adversarial network, in: Artificial Intelligence Research and Development: Proceedings of the 23rd International Conference of the Catalan Association for Artificial Intelligence, 2021, p. 392.
[17] M.M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, P. Vandergheynst, Geometric deep learning: going beyond euclidean data, IEEE Signal Processing Magazine 34 (4) (2017) 18–42.
[18] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).
[19] J. Liang, Y. Deng, D. Zeng, A deep neural network combined cnn and gcn for remote sensing scene classification, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 13 (2020) 4325–4338.
[20] L. Zhang, X. Li, A. Arnab, K. Yang, Y. Tong, P.H. Torr, Dual graph convolutional network for semantic segmentation, arXiv preprint arXiv:1909.06121 (2019).
[21] J. Fu, J. Liang, Z. Wang, Monocular depth estimation based on multi-scale graph convolution networks, IEEE Access 8 (2019) 997–1009.
[22] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.
[23] H. Zhao, O. Gallo, I. Frosio, J. Kautz, Loss functions for image restoration with neural networks, IEEE Transactions on computational imaging 3 (1) (2016) 47–57.
[24] C. Shu, K. Yu, Z. Duan, K. Yang, Feature-metric loss for self-supervised learning of depth and egomotion, European Conference on Computer Vision, Springer (2020) 572–588.
[25] H. Fu, M. Gong, C. Wang, K. Batmanghelich, D. Tao, Deep ordinal regression network for monocular depth estimation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2002–2011.
[26] I. Alhashim, P. Wonka, High quality monocular depth estimation via transfer learning, arXiv preprint arXiv:1812.11941 (2018).
[27] Y. Chen, H. Zhao, Z. Hu, J. Peng, Attention-based context aggregation network for monocular depth estimation, International Journal of Machine Learning and Cybernetics 12 (6) (2021) 1583–1596.
[28] V.M. Babu, K. Das, A. Majumdar, S. Kumar, Undemon: Unsupervised deep network for depth and ego-motion estimation, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1082–1088.
[29] M. Zhang, X. Ye, X. Fan, W. Zhong, Unsupervised depth estimation from monocular videos with hybrid geometric-refined loss and contextual attention, Neurocomputing 379 (2020) 250–261.
[30] A. Masoumian, H.A. Rashwan, J. Cristiano, M.S. Asif, D. Puig, Monocular depth estimation using deep learning: A review, Sensors 22 (14) (2022) 5353.
[31] Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, Lego: Learning edge with geometry all at once by watching videos, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 225–234.
[32] Z. Lei, Y. Wang, Z. Li, J. Yang, Attention based multilayer feature fusion convolutional neural network for unsupervised monocular depth estimation, Neurocomputing 423 (2021) 343–352.
[33] H. Wang, Y. Sun, Q.J. Wu, X. Lu, X. Wang, Z. Zhang, Self-supervised monocular depth estimation with direct methods, Neurocomputing 421 (2021) 340–348.
[34] L. He, J. Lu, G. Wang, S. Song, J. Zhou, Sosd-net: Joint semantic object segmentation and depth estimation from monocular images, Neurocomputing 440 (2021) 251–263.
[35] L. Liu, X. Song, M. Wang, Y. Liu, L. Zhang, Self-supervised monocular depth estimation for all day images using domain separation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12737–12746.
[36] T. Zhou, M. Brown, N. Snavely, D.G. Lowe, Unsupervised learning of depth and ego-motion from video, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1851–1858.
[37] Z. Yin, J. Shi, Geonet: Unsupervised learning of dense depth, optical flow and camera pose, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 1983–1992.
[38] C. Wang, J.M. Buenaposada, R. Zhu, S. Lucey, Learning depth from monocular videos using direct methods, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 2022–2030.
[39] R. Mahjourian, M. Wicke, A. Angelova, Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 5667–5675.
[40] A. Gordon, H. Li, R. Jonschkowski, A. Angelova, Depth from videos in the wild: Unsupervised monocular depth learning from unknown cameras, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 8977–8986.
[41] H. Singh, R. Sharma, Role of adjacency matrix & adjacency list in graph theory, International Journal of Computers & Technology 3 (1) (2012) 179–183.
[42] J.B. Estrach, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and deep locally connected networks on graphs, in: 2nd international conference on learning representations, ICLR, Vol. 2014, 2014.
[43] H.A. Rashwan, S. Chambon, P. Gurdjos, G. Morin, V. Charvillat, Using curvilinear features in focus for registering a single image to a 3d object, IEEE Transactions on Image Processing 28 (9) (2019) 4429–4443.
[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, Automatic differentiation in pytorch (2017).
[45] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[46] M. Tan, Q. Le, Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR, 2019, pp. 6105–6114.
[47] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: 2012 IEEE conference on computer vision and pattern recognition, IEEE, 2012, pp. 3354–3361.
[48] A. Saxena, M. Sun, A.Y. Ng, Make3d: Learning 3d scene structure from a single still image, IEEE transactions on pattern analysis and machine intelligence 31 (5) (2008) 824–840.
[49] D. Eigen, R. Fergus, Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 2650–2658.
[50] Z. Yang, P. Wang, W. Xu, L. Zhao, R. Nevatia, Unsupervised learning of geometry from videos with edge-aware depth-normal consistency, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018.
[51] Y. Zou, Z. Luo, J.-B. Huang, Df-net: Unsupervised joint learning of depth and flow using cross-task consistency, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 36–53.
[52] C. Luo, Z. Yang, P. Wang, Y. Wang, W. Xu, R. Nevatia, A. Yuille, Every pixel counts++: Joint learning of geometry and motion with 3d holistic

understanding, IEEE transactions on pattern analysis and machine intelligence 42 (10) (2019) 2624–2641.
[53] V. Casser, S. Pirk, R. Mahjourian, A. Angelova, Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 33, 2019, pp. 8001–8008.
[54] Y. Meng, Y. Lu, A. Raj, S. Sunarjo, R. Guo, T. Javidi, G. Bansal, D. Bharadia, Signet: Semantic instance aided unsupervised 3d geometry perception, in: Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition, 2019, pp. 9810–9820.
[55] A. Ranjan, V. Jampani, L. Balles, K. Kim, D. Sun, J. Wulff, M.J. Black, Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 12240–12249.
[56] J. Zhou, Y. Wang, K. Qin, W. Zeng, Unsupervised high-resolution depth learning from videos with dual networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6872–6881.
[57] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, A. Gaidon, 3d packing for self-supervised monocular depth estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2485–2494.
[58] U.-H. Kim, S.-H. Kim, J.-H. Kim, Simvodis: Simultaneous visual odometry, object detection, and instance segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (1) (2020) 428–441.
[59] L. Liebel, M. Körner, Multidepth: Single-image depth estimation via multi-task regression and classification, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), IEEE, 2019, pp. 1440–1447.
[60] M. Goldman, T. Hassner, S. Avidan, Learn stereo, infer mono: Siamese networks for self-supervised, monocular, depth estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2019.
[61] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, N. Navab, Deeper depth prediction with fully convolutional residual networks, in: 2016 Fourth international conference on 3D vision (3DV), IEEE, 2016, pp. 239–248.
[62] K. Karsch, C. Liu, S.B. Kang, Depth transfer: Depth extraction from video using non-parametric sampling, IEEE transactions on pattern analysis and machine intelligence 36 (11) (2014) 2144–2158.
[63] M. Liu, M. Salzmann, X. He, Discrete-continuous depth estimation from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 716–723.

Armin Masoumian received the B.Sc. degree in mechatronics engineering from the University of Debrecen, Debrecen, Hungary, in 2017 and the M.Sc. degree in mechatronics systems from Kingston University, London, U.K. He is currently pursuing the Ph.D. degree with the IRCV Group at University of Rovira i Virgili (URV), Tarragona, Spain. Also, He is working as a scholar visitor at University of California, Riverside, CA, USA. His current research interests include machine learning, deep learning, computer vision, robotics and mechatronics.

Hatem A. Rashwan received the B.S. and M.S. degrees in electrical engineering from South Valley University, Egypt, in 2002 and 2007, respectively, and the Ph.D. degree in computer vision from Universitat Rovira i Virgili, Spain, in 2014. From 2004 to 2009, he joined the Electrical Engineering Department, South Valley University, as an Assistant Lecturer. From January 2010 until October 2014, he joined the IRCV Group, Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, as a Research Assistant. From November 2014 until August 2017, he was a Post-Doctoral Researcher with the VORTEX Group, IRIT, CNRS, INP-Toulouse, University of Toulouse, France. Since 2018, he has been a Beatriu de Pinos Researcher with URV. His research' interests include image processing, computer vision, machine learning, and pattern recognition.

Saddam Abdulwahab received the B.S. degree in computer science from Hodeidah University, Hodeidah, Yemen, in 2012, and the M.Sc. degree in computer security and artificial intelligence from URV, Tarragona, Spain, in 2017. He is currently pursuing the Ph.D. degree with the IRCV Group. From 2012 to 2016, he joined the Department of Computer Science and Engineering, Hodeidah University, as a Lecturer. In 2016, he joined the Intelligent Technologies for Advanced Knowledge Acquisition ITAKA Group, DEIM, URV. His research interests include image processing, computer vision, machine learning, and pattern recognition.

Julian Cristiano' received the B.S degree in Electronic Engineering from the Industrial University of Santander in 2007, Bucaramanga, Colombia and the M.S.and Ph.D. degrees in Computer Science from Rovira i Virgili University, Tarragona, Spain in 2009 and 2016, respectively. He is currently senior postdoctoral researcher at the intelligent robotics and computer vision group (IRCV). His research interests include artificial intelligence, robotics, biologically inspired control and evolutionary computation.

M. Salman Asif received the B.Sc. degree from the University of Engineering and Technology, Lahore, Pakistan, and the M.S. and Ph.D. degrees from the Georgia Institute of Technology, Atlanta, GA, USA. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, University of California, Riverside, CA, USA. He was a Senior Research Engineer with Samsung Research America, Dallas, TX, USA from 2012 to 2014, and a Postdoctoral Researcher with Rice University, Houston, TX, from 2014 to 2016. His research interests include computational imaging, signal/image processing, computer vision, and machine learning. He was the recipient of Hershel M. Rich Outstanding Invention Award in 2016, Google Faculty Award in 2019, and NSF CAREER Award in 2021.

Domenec Puig received the M.S. and Ph.D. degrees in computer science from the Polytechnic University of Catalonia, Barcelona, Spain, in 1992 and 2004, respectively. In 1992, he joined the Department of Computer Engineering and Mathematics, Universitat Rovira i Virgili, Tarragona, Spain, where he is currently a Professor. Since July 2006, he has been the Head of the Intelligent Robotics and Computer Vision Group, Universitat Rovira i Virgili. His research interests include image processing, texture analysis, perceptual models for image analysis, scene analysis, and mobile robotics.