ISPRS Journal of Photogrammetry and Remote Sensing 176 (2021) 1–14

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Self-supervised monocular depth estimation from oblique UAV videos

Logambal Madhuanand, Francesco Nex, Michael Ying Yang *

Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, Enschede, the Netherlands

ARTICLE INFO

Keywords:
Depth estimation
Monocular
UAV video
Self-supervised learning
Scene Understanding

ABSTRACT

Unmanned Aerial Vehicles (UAVs) have become an essential photogrammetric measurement as they are affordable, easily accessible and versatile. Aerial images captured from UAVs have applications in small and large scale texture mapping, 3D modelling, object detection tasks, Digital Terrain Model (DTM) and Digital Surface Model (DSM) generation etc. Photogrammetric techniques are routinely used for 3D reconstruction from UAV images where multiple images of the same scene are acquired. Developments in computer vision and deep learning techniques have made Single Image Depth Estimation (SIDE) a field of intense research. Using SIDE techniques on UAV images can overcome the need for multiple images for 3D reconstruction. This paper aims to estimate depth from a single UAV aerial image using deep learning. We follow a self-supervised learning approach, Self-Supervised Monocular Depth Estimation (SMDE), which does not need ground truth depth or any extra information other than images for learning to estimate depth. Monocular video frames are used for training the deep learning model which learns depth and pose information jointly through two different networks, one each for depth and pose. The predicted depth and pose are used to reconstruct one image from the viewpoint of another image utilising the temporal information from videos. We propose a novel architecture with two 2D Convolutional Neural Network (CNN) encoders and a 3D CNN decoder for extracting information from consecutive temporal frames. A contrastive loss term is introduced for improving the quality of image generation. Our experiments are carried out on the public UAVid video dataset. The experimental results demonstrate that our model outperforms the state-of-the-art methods in estimating the depths.

1. Introduction

Unmanned Aerial Vehicles (UAVs) are commonly used photogrammetric measurement platforms that gained popularity due to its accessibility, affordability and its versatility in capturing images. The images captured from UAVs can be used for applications like Digital Surface Model (DSM) generation, Digital Terrain Model (DTM) extraction, mapping, textured 3D models, etc. (Nex and Remondino, 2014). The acquisition of images by UAVs can either be through nadir view or oblique view based on its application. The images acquired from nadir view have a constant scale and good coverage whereas oblique view images have wider coverage, provides higher redundancy, better 3D model formation and can be also useful for an improved orthophoto generation (Nex et al., 2015). Yet, oblique images have a varying scale, occlusions and illumination problems which requires detailed processing (Aicardi et al., 2016). A set of oblique view images with sufficient overlap, acquired from UAVs can be used for reconstructing a dense 3D model through photogrammetric image techniques like Structure-from-Motion (SfM). The general steps involve feature detection and matching by finding correspondences between images, sparse reconstruction, bundle adjustment followed by 3D dense point clouds generation (Vallet et al., 2012; Voumard et al., 2018; Nex et al., 2015). However, these imaging techniques suffer from issues like requirement of multiple images of the same scene, time consumption, low-quality generation in regions where the camera has a narrow view and for images captured along camera optical axis. This led to an increased interest towards image-based 3D reconstruction techniques that can overcome these problems.

Recent advancements in computer vision and deep learning techniques have increased the focus towards depth estimation by using single images, doing away with the need for multiple images (Eigen et al., 2014; Liu et al., 2015; Garg et al., 2016; Godard et al., 2017). Single Image Depth Estimation (SIDE) models use depth cues like contextual information, texture variations, gradients, shading, defocus etc., for obtaining depth images from a single (monocular) image (Saxena et al., 2005). A depth image represents the distance from the

* Corresponding author.
E-mail addresses: logambal.ceg@gmail.com (L. Madhuanand), f.nex@utwente.nl (F. Nex), michael.yang@utwente.nl (M.Y. Yang).

https://doi.org/10.1016/j.isprsjprs.2021.03.024
Received 2 December 2020; Received in revised form 28 March 2021; Accepted 29 March 2021
Available online 13 April 2021

viewpoint to the scene object at a particular orientation in each pixel. These depth images have applications in 3D reconstruction, semantic segmentation, object identification and tracking, monitoring topographic changes, etc. (Chen et al., 2018). Most of the studies that deal with single image depth, use Convolutional Neural Networks (CNN) as CNNs have proven capable of learning fine details by combining low level and high level features (Bhandare et al., 2016) that are beneficial for this extraction task.

Nearly all previous works dealing with SIDE models have been focused on images taken at ground level or indoor images while monocular depth estimation from aerial images has been attempted very rarely. Compared to terrestrial images, aerial images may lack information like shading, texture etc., making it more difficult for the learning process. Also its varied perspectives, increased distance between the camera and objects in the scene along with its larger area of coverage makes it a challenging task to be addressed. Single image depth from UAV aerial images can have many applications, such as, for preparing low cost elevation models, for 3D reconstruction with a minimal number of images captured beforehand for reconstruction, damage evaluation in places where regular photogrammetric block acquisition is not possible, on-board UAVs for augmented Simultaneous Localization And Mapping (SLAM) to estimate the position of vehicles etc. The depth from single images can be back projected to form point clouds and merged in regions with sparse point clouds instead of interpolation. Comparatively single depth images are cheaper to acquire than other sources of active depth estimation devices. Also, depths from single UAV images can be used for collision avoidance, object with several occlusions are captured, identifying the object positions and to fill holes and gaps in the point cloud. Representative monocular depth estimated from single UAV images in UAVid dataset (Lyu et al., 2020) is shown in Fig. 1.

Monocular depth using deep learning models have been commonly approached by training the CNN network with corresponding ground truth depth maps (Eigen et al., 2014; Liu et al., 2015; Laina et al., 2016; Li et al., 2017; Mou and Zhu, 2018; Koch et al., 2019; Amirkolaee and Arefi, 2019). Even though the predicted depth maps from supervised SIDE models look very similar to the ground truth depth, collecting large quantities of pixel-wise ground truth depth for training is a tiresome process. This led towards the development of Self-Supervised Monocular Depth Estimation (SMDE) techniques. SMDE techniques learn depth from complementary information with depth results comparable in accuracy to that of supervised depth learning model. Available SMDE techniques learn depth through either stereo pairs or monocular videos. When using stereopairs, the SMDE model takes one image from the pair as input to the deep learning model to predict disparity. The predicted disparity is warped with the other image from the pair to generate the original input image. The difference between the generated and original image is backpropagated for improving the model performance. The produced disparities can be converted to depth using the baseline and focal length which is usually done as an offline process. However, preparation of stereopairs dataset from UAV aerial images is a challenging task and the model trained with stereopairs might have issues due to occlusions (Godard et al., 2017). While using monocular videos for training, the SMDE model learns both depth and pose information jointly through two different networks, one each for depth and pose. The predicted depth and pose are used to reconstruct one image from the viewpoint of another image utilising the temporal information from videos. Acquiring videos are becoming increasingly popular as they offer flexibility during data collection. With the developments in instruments, it has become possible to acquire videos at resolutions that are comparable with high quality still images. As most of the UAV platforms captures videos, depth estimation from monocular videos through SMDE is preferred.

In this paper, we propose an architectural improvement and an additional loss term for enhancing the qualitative and quantitative performance of monocular depth model estimation from oblique UAV videos. The contributions of this work are the following. (1) To the best of our knowledge, we present the first work of self-supervised monocular depth estimation model for oblique UAV videos. (2) Technically, we use two consecutive temporal frames in two 2D CNN encoders to form feature maps that are given as an input to 3D CNN depth decoders for generating inverse depths. (3) Furthermore, we propose to add a contrastive loss term in the training phase to make the model produce images closer to the original video images. The code is publicly available at https://github.com/logsanand/Monocular-UAV-videos.

This paper is organised as follows. Section 2 gives an outline about the related work similar to our work. Section 3 presents the structure of the proposed model architecture. The experimental implementation and the results are explained in Section 4, which is followed by conclusions.

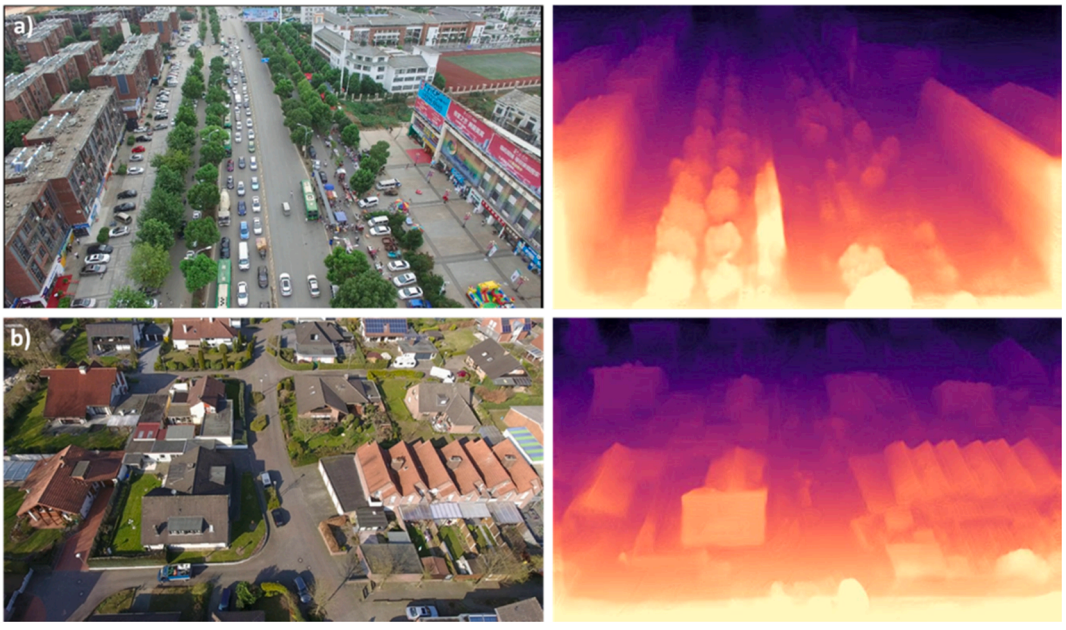

Fig. 1. Disparity from two representative monocular images in UAVid dataset (Lyu et al., 2020) a) Wuhan, China, B) Gronau, Germany.

2. Related work

Depth estimation from images using computer vision techniques is very popular due to its successful performance with terrestrial images. It includes the use of multiple image views of the same scene (Szeliski & Zabih, 2000; Remondino et al., 2013; Furukawa and Hernández, 2013), stereopairs (Alagoz, 2016), illumination or texture cues (Zhang et al., 1999) etc., to reconstruct the 3D model. Extraction of 3D information from single images relied on techniques like shape from shading (Van Den Heuvel, 1998), shape from texture (Kanatani and Chou, 1989), followed by the use of stereo or temporal sequences from 2D images. Based on the additional constraints used for extracting depth from single images, they are clustered into different categories. Li et al., (2020) reviewed various SIDE models and datasets along with the applications benefitted from these models. The SIDE models explained below are some of the important studies which contributed to the development of our model and are mostly based on SIDE models that are applied to aerial/UAV images.

2.1. Depth from stereo images

Binocular stereo or multi-baseline stereo (Kang et al., 1999) techniques that are used to extract 3D information from two or multiple images covering the same scene forms the traditional approach. This involves matching pixels across multiple rectified images and using them to orient the images for the point cloud generation. The distance between these corresponding points in the left and right image defines the disparity map of the images. This disparity map can be used for 3D reconstruction. The disparity and depth information are related inversely as $Disparity = X_l - X_r = \frac{Bf}{d}$, where X_l and X_r denote the corresponding image points, B represents the baseline distance between cameras, f is the camera constant and d is the depth or object distance from the viewpoint. The obtained disparity map can be used to calculate the depth information from the images through the baseline and camera constant. This process of finding the difference between corresponding points from two or more viewpoints forms the base of stereo matching algorithms. Stereo matching algorithms include estimating the matching cost, cost aggregation, disparity calculation and optimization using low level image features to measure dissimilarity (Zhou et al., 2020). One among them is Semi-Global Matching (SGM) proposed by Hirschmuller (2008) that has wider adoption in many recent computer vision tasks due to the quality results and its faster performance. SGM is a dense image matching technique, which matches pixel-wise mutual information by matching cost. Instead of using intensity difference alone for matching, SGM uses disparity information to find the corresponding pixels in other images. Deep learning has been successfully used to overcome many of the difficulties in traditional stereo algorithms. Žbontar and Le Cun, (2015) used CNN to compute matching cost and depth from the rectified image pair. Mayer et al., (2016) used FlowNet to estimate both disparity and scene flow with large scale datasets. Kendall et al., (2017) employed 3D convolutions using GC-NET for regularizing cost volume and regressing disparity from stereo images. Liang et al., (2018) proposed a network architecture to include all steps of stereo matching algorithms by using a different disparity refinement network for joint learning. There are also other deep learning approaches which produce highly accurate depths from stereo image pairs. Using a stereo device to capture stereo images or to create a stereo image dataset from overlapped images is a relatively cumbersome process as compared to obtaining videos. Occasionally, the overlap might not be sufficient leading to potential use of single image depth estimation models. Obviously using stereo images compared to single images has its advantage of viewing the same scene from two angles. Many of these methods acted as a base for single image depth estimation techniques where the limitation of acquiring multiple images of the same scene can be overcome.

2.2. Single image depth estimation with supervised learning

Depth from single images has been in existence for a long time and has been implemented by imposing specific conditions or adding complementary information. Saxena et al., (2005) used Markov Random Fields (MRF) to combine local and global features to produce depth using the contextual relationship between features while avoiding occlusions. With recent developments in computer vision and deep learning techniques, there is an increasing possibility of overcoming the limitations of analytical methods. This is mainly due to the success of Convolutional Neural Networks (CNN) in learning depth from colour intensity images. Several studies have been published on depth estimation from a single image using ground truth depths for training deep learning models. Most of the studies have been on indoor images or outdoor images taken at ground level. Only a few studies have analysed the techniques for depth estimation from aerial images. Julian et al., (2017) compared different style transfer methods like pix2pix, cycle GAN and multi-scale deep network for aerial images captured from UAVs. They trained the model using the UAV images along with depth image pairs and refined the feature-based transfer algorithm for this single image depth estimation purpose. Mou & Zhu, (2018) used a fully residual convolutional-deconvolutional network for extracting depth from monocular imagery. They used aerial images along with the corresponding DSM generated through semi-global matching for training the network. The two parts of the networks acts as a feature extractor and height generator. Amirkolaee & Arefi, (2019) proposed a deep CNN architecture with an encoder-decoder setup for estimating height from aerial images by training them with the corresponding DSM. They extracted the full satellite image into local patches and trained the model with the corresponding depth and finally stitched the depths together. They faced issues for small objects with fewer depth variations like small vegetation, ground surfaces within the scene etc. Although all these methods proved to be successful, they all require huge amounts of ground truth depth images while training the model. Preparing large amounts of UAV images along with their corresponding DSM is complicated, making supervised approach less preferable compared to other approaches even though it produces slightly better accuracies for single image depth estimation.

2.3. Single image depth estimation with self-supervised learning using stereopairs

The limitation in acquiring large ground truths for training led towards the growth of studies towards SMDE techniques. The reduced dependencies on laborious ground truth data collection have generated a lot of interest in these approaches. Garg, Vijay Kumar, Carneiro, & Reid, (2016) circumvent the problem faced by supervised learning by utilising stereo images instead of ground truth depth maps. They used the 3D reconstruction concept to generate a disparity image from stereo images and reconstruct the original image through inverse warping. They suggested this approach can be continuously updated with data and fine-tuned for specific purposes. Although the model performed well, their process of generating images is not fully differentiable due to the non-linear loss terms. The use of Taylor approximation makes the loss terms more complex. Godard et al., (2017) overcame this by including a fully differential training loss term for left–right consistency of the generated disparity image to improve the quality of the generated depth image. The adversarial learning models mark the current state of the art in many areas where deep learning is being used. A simple GAN network consists of a generator that learns to produce realistic images and discriminator that learns to find the difference with real images. MonoGAN by Aleotti, Tosi, Poggi, & Mattoccia, (2018) used a combination of generator and discriminator network for the monocular depth estimation. Mehta et al. (2018) used this structured adversarial training to improve the task of image reconstruction for predicting depth images from the stereo images. Tosi et al., (2019) used traditional stereo

knowledge along with the combination of synthesised depths for estimation of depth from monocular images. Madhuanand et al., (2020) used stereo images for depth estimation from single UAV images using two deep learning models and compared the model performances. These SMDE models use stereo pairs for replacing the ground truth information while training.

2.4. Single image depth estimation with self-supervised learning from videos

UAV videos are easier to acquire than producing stereo pairs though there are additional complexities in a UAV video-based SMDE model as it needs to estimate both depth and position. Most of the studies using videos for self-supervised monocular depth estimation are carried out on terrestrial images. Some of the important studies which helped to formulate our architecture have been discussed below. Zhou et al., (2017) proposed a novel framework to jointly learn depth and ego-motion with videos using adjacent temporal frames in a self-supervised manner. Similarly, Vijayanarasimhan et al., (2017) used SfM-Net for geometry aware depth estimation for certain objects in a scene with an option available for making it a supervised approach. Following this, different approaches for estimating depth from monocular videos have been proposed. Mahjourian et al., (2018) used 3D geometrical constraints for improving the monocular depth estimation performance. Other works which have modifications in depth estimation network or additional loss terms for increased performance includes Poggi et al., (2018), Wang et al., (2018), Godard et al., (2019), Dai et al., (2019), Tan et al., (2020), Spencer et al., (2020), Guizilini et al., (2020) etc. These models are trained and tested only with terrestrial videos and their performance on UAV videos may not be similar as UAV scene has varied perspectives, much higher range of depths within the scene and has larger area of coverage. Hermann et al., (2020) used a similar architecture to Zhou et al., (2017) and trained it on monocular UAV videos for a self-supervised monocular depth estimation task. The architecture jointly estimates depth and pose for simultaneous learning and prediction. The model takes three consecutive frames for pose estimation and a single frame for depth estimation to reconstruct the input image. They adopted image reconstruction loss, smoothing loss and edge-aware loss, for improving the performance. Their study separately trained and tested three different datasets - rural, urban and synthetic dataset, and evaluated the model performance. The depths generated by Hermann et al., (2020) over rural region is quantitatively better than that over urban region as the increased complexity of the urban scene proved challenging for the model to resolve. However their rural dataset is simplistic and generally most of the real world scenes will be more complicated and have lot more objects than depicted in the rural dataset of Hermann et al., (2020). Also, in order for such models to be implemented in different regions, it is important to understand the transferability of the model over various scenes which is not studied by Hermann et al., (2020). We aim to implement a model that can perform well in a complicated UAV scene similar to real time videos from urban environments. We modify the architecture, experiment with different combinations of loss terms and study the transferability of the model to understand its ability to perform in complex regions.

3. Method

This section describes the overall methodology of the self-supervised depth estimation model from UAV oblique videos. This involves the preparation of input images to the network, training the deep learning model and fine-tuning the model for hyper-parameters. The model predicts depth map using depth network and then estimates rotation and translation parameters from consecutive temporal frames using the pose network which are used to reconstruct the given input image. The difference between the reconstructed and the original input image is calculated as a loss to be back-propagated for improving the model performance. The quality of the estimated single depth images is evaluated using various statistics by comparing with ground truth or reference depth images produced using Pix4D ("Pix4D (version 4.4.12)," 2020). The architectural outline of the methodology is shown in Fig. 2.

The architecture includes two successive temporal frames used as an input to two 2D encoders which are extracted as depth through a 3D depth decoder. This predicted depth along with pose information using three temporal frames is used to reconstruct the input image. The difference between the reference and the original input image is used as a loss to improve the model performance.

3.1. Input to network

The SMDE model requires a large amount of training images. The training images are extracted from temporal frames of UAV videos. The training involves sets of three consecutive temporal images that are provided as an input to the model. The three images are chosen such that the baseline between them is maintained without disturbing the image reconstruction process. The three temporally adjacent images are referred as I_{r-1}, I_r, I_{r+1}, where I_r, represents the reference image, I_{r-1} and I_{r+1} represents the source images at a previous frame and image at a subsequent frame respectively. The camera intrinsic parameters which include principal point location and focal length information are required to estimate the pose information.

3.2. Network architecture

The SMDE model consists of two networks, one each for estimating depth and pose information from temporal image sequences in a self-supervised manner. The depth network takes as input two image frames (reference and subsequent source image) which passes through the encoder-decoder structure to estimate inverse disparity through the image reconstruction process. The pose network takes as input the reference image and the two source images to estimate the translation and orientation parameters. The predicted pose and depth image along with the source images is used to reconstruct the reference image. This is achieved by taking Depth image (D) generated from depth network, camera transformation matrix ($P_{r \rightarrow r'}$) from pose network and a source image as input. It uses bilinear sampling by sampling pixels for projecting the source view with depth and pose information to reconstruct the reference image as shown in Eq. (1). This process of reconstructing the reference image is based on (Godard et al., 2019).

$$I_{r' \rightarrow r} = I_{r'}[proj(D_r, P_{r \rightarrow r'}, K)] \quad (1)$$

where $I_{r' \rightarrow r}$ represent the reconstructed image by the projection of source image into reference image, $I_{r'}$ represents the source image (both I_{r-1} previous and I_{r+1} subsequent frames), D_r represents the predicted depth, $P_{r \rightarrow r'}$ represents the relative camera pose between the reference and source image, K represents the camera intrinsic parameters.

3.2.1. Depth network

For the depth estimation task, we follow the encoder-decoder network architecture similar to the one proposed by Godard et al., (2019) and have modified the layers and structure of the encoder and decoder network to optimize for the performance. In order to extract more information from the input images, we propose two encoders constructed with ResNet18 (He et al., 2016) architecture. The encoders use pre-trained weights to guide the model towards global minimum. The two encoders take as input the reference image (I_r) and the successive temporal source image (I_{r+1}). They pass through the encoder with multiple skip connection to extract both low level and high level features. Using two image frames instead of one can help in extracting more features and can also be used for building occluded objects in one frame with the other. The two images are passed through the first layer with a kernel size of 7x7, followed by 3x3 filter with an increasing

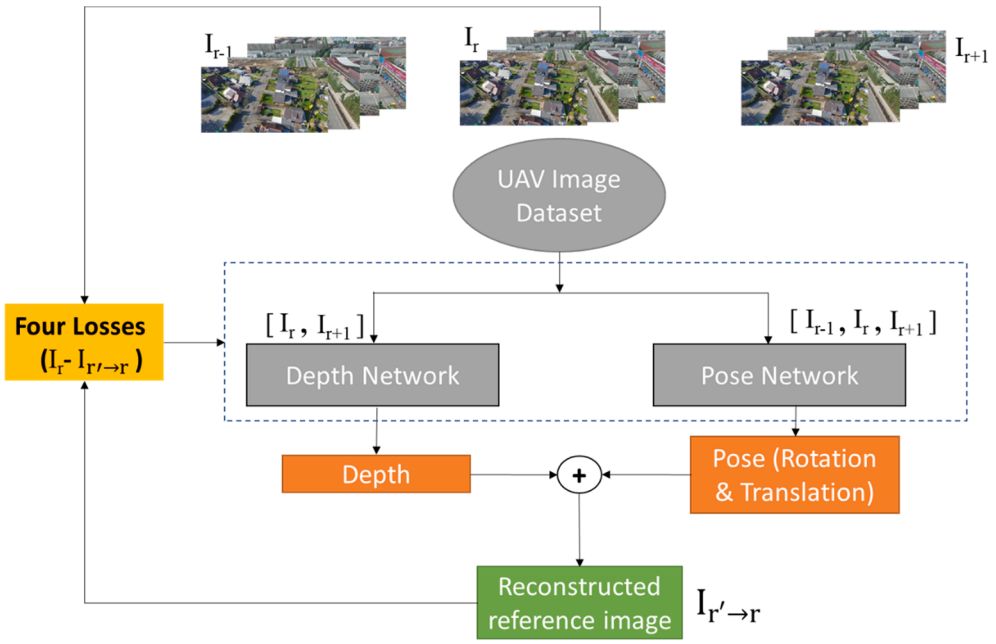

Fig. 2. Network architecture with Depth network and Pose network taking input from the UAV video dataset.

number of features at each layer. The outputs from the two encoders are concatenated to form an extra dimension. The depth decoder is formed with a 3D convolutional network that uses the volumetric information from both features. This 3D CNN can help in extracting features from temporal sequences through convolutional blocks. This is followed by up-sampling layers to match the layers as that of the original image size. The features from all the layers are averaged to form the last layer which consists of sigmoid activation to predict the Depth (D) map. The network architecture is explained in Fig. 3.

3.2.2. *Pose network*

The network architecture of the encoder for pose network uses the same number of layers as depth encoder and is given in Fig. 4. The reference and two source images are given as input to the pose encoder. The feature maps are generated from the ResNet-18 encoder network. The generated feature maps from reference and one source image are concatenated to form the dense layers. These feature maps are passed through pose decoder. The features are passed through four convolutional layers of 256 features each with 3×3 kernel size. These layers are concatenated together to form a feature map with 6 degrees of freedom to be predicted from a pair of images. This is passed through ReLU activation function and mean of these features are taken. These features are split into three layers for axis angle and three layers for translation. Thus, the pose is estimated from the two source images (I_{r-1}, I_{r+1}) with respect to the reference image (I_r) and is denoted as $P_{r'}(P_{r\to(r+1)}, P_{r\to(r-1)})$. It is composed of a 4×4 camera transformation matrix with rotation and translation parameters between source view and reference view. This pose information along with predicted depth and camera matrix is used to reconstruct the reference image as given in Eq. (1).

3.3. *Loss functions*

To increase the model performance and improve the generated depth quality, various loss terms are used during training. Unlike supervised

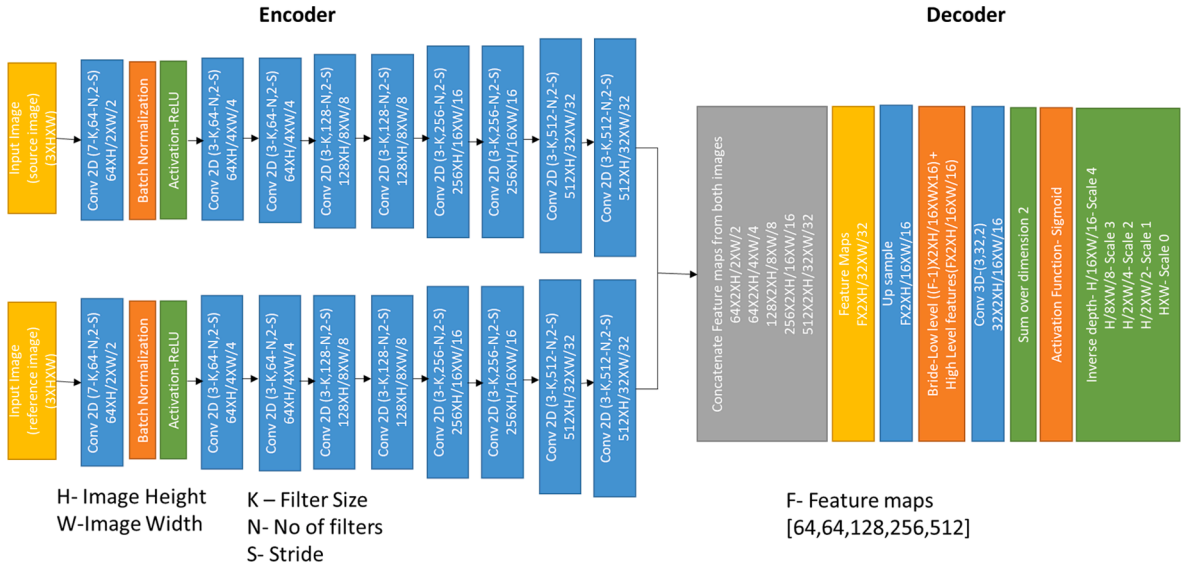

Fig. 3. Depth Network architecture Encoder -Decoder with number of layers.

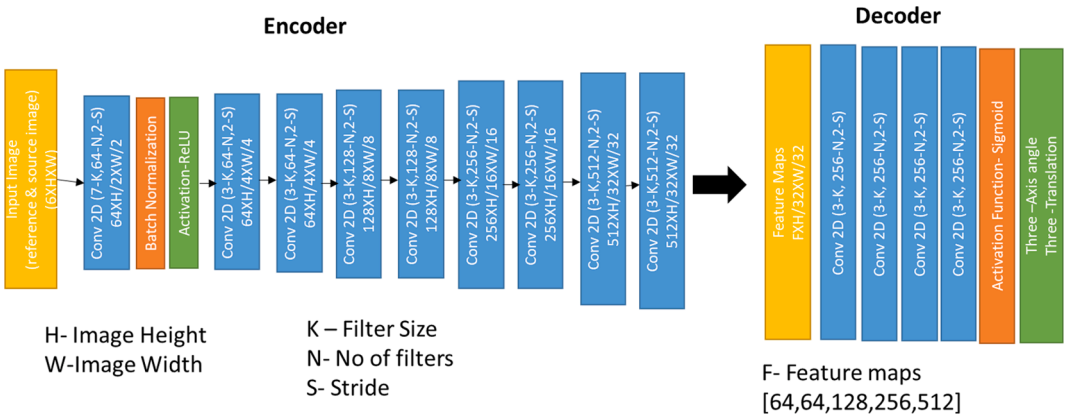

Fig. 4. Pose Network architecture Encoder -Decoder with number of layers.

techniques which use the ground truth depth to increase the predicted depth, we use the generated reference image for backpropagation to increase the quality of generated depth. This is handled through different loss terms as described in Eq. (2).

The total loss is given by

$$L = \lambda_1 L_p + \lambda_2 L_S + \lambda_3 L_M + \lambda_4 L_C \tag{2}$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ represents the weights between different loss terms used. L_p, represents the reprojection loss, L_S, represents the smoothing loss, L_M, represents the masking loss and L_C, represents the contrastive loss term. The importance of different loss terms and the selection of weights are discussed in Section 4.

3.3.1. Reprojection loss

The difference between the reconstructed reference image and the original image is minimized using the photometric loss such that the depth quality increases. This is a combination of both L1 and SSIM loss (Wang et al., 2004) for identifying the difference as shown in Eq. (3).

$$L_p = \frac{1}{N}\sum \alpha \frac{(1 - SSIM(I_a, I_b)}{2} + (1-\alpha)||I_a - I_b|| \tag{3}$$

where α = 0.85, $I_a$and I_b represents the reference image and the reconstructed reference image. Godard et al., (2019) suggested the use of minimum pixel reprojection loss instead of the average reprojection loss due to its minimisation of artefacts near image borders. For our case, we tested both average and minimum reprojection loss and found that minimum reprojection loss is found to be more suitable for our objective.

3.3.2. Smoothness loss

To have a smoother depth images, edge-aware smoothness loss is implemented following Wang et al.,(2018). Due to discontinuities in depth which is reflected in the colour gradient of the image, we also used this loss to produce a normalized depth map. This is given in Eq. (4)

$$L_S = \left|\frac{\partial_x d}{d^-}\right| e^{-|\partial_x I_r|} + |\partial_y d/d^-|e^{-|\partial_y I_r|} \tag{4}$$

where d/d^- is the normalised depth, the gradient of normalised depth is weighted with the colour gradient of the reference image I_r in x and y directions.

3.3.3. Masking loss for dynamic objects

The dataset we use for training has a significant number of frames with dynamic objects like moving cars. Self-supervised monocular depth estimation, however, works with the assumption that only the camera is moving while the objects in the scene do not move or the camera is static. Godard et al., (2019) suggested a masking loss, in order not to consider the dynamic objects in the static scenes during loss calculation. A similar loss is implemented in our network, as our dataset has many moving cars and static scenes making it more representative of real life situation, compared to standard datasets like KITTI (Geiger et al., 2012). The loss is calculated per-pixel wise, where the reprojection error for the reconstructed image and reference image is compared with the reprojection error between reference and source images. Only those pixels where the reprojection error between the reconstructed image and the reference image is less than the reprojection error between the reference image and the source image is considered for error calculation by applying the mask, $\boldsymbol{L_M}$ as given in Eq. (5),

$$L_M = min[L_p(I_r, I_{r'\to r})] < min[L_p(I_r, I_{r'})] \tag{5}$$

where $I_{r'\to r}$ represents the reconstructed image, I_r represents the reference image, $I_{r'}$ represents the source image and $\boldsymbol{L_p}$ represents the reprojection error.

3.3.4. Contrastive loss

In order to improve the quality of the generated images, we add a contrastive loss term as suggested by Spencer et al., (2020). They have proposed the contrastive loss term for improving the feature map generation process. We have modified this slightly and used it to compared the loss between reconstructed reference image and original input image. It takes feature vectors from the reference image I_r and reconstructed image $I_{r'\to r}$, calculates the distance between the two and is defined as given in Eq. (6),

$$l(y, r_1, r_2) = \begin{cases} \frac{1}{2}d^2 if(y=1) \\ \frac{1}{2}\{max(0, m-d)\}^2 if(y=0) \\ 0 otherwise \end{cases} \tag{6}$$

where, y indicates whether the pairs are positive or negative correspondence (1,0), d=$||r_1 - r_2||$ is the Euclidean distance calculated between the pairs, m is the margin set between positive and negative pairs. A positive pair is the one that has a distance between feature vectors low while negative pairs contribute to larger distances between the pairs. Here the pairs refer to the reference image and reconstructed reference image. This is shown in Eq. (7)

$$L_C = \Sigma l(y, I_r, I_{r'\to r}) \tag{7}$$

The total loss is included based on whether the pairs are positive or negative or should be ignored. This loss serves in matching the features despite the appearance changes.

3.4. Inference

As our study is to estimate depth from monocular UAV videos, our

main focus is to improve the depth estimation model. Hence, our objective is oriented towards modifying the depth network architecture. Compared to Godard et al., (2019)'s model, we added another encoder to extract more information from consecutive frames and to optimize the image information from this volumetric feature maps, a 3D depth decoder is also added. In addition to this, we included a contrastive loss term in our network to improve the image generation process through feature matching. The depth network is modified to improve the quality of the generated disparity maps and also to produce depth maps closer to the ground truth. To assess the performance of the proposed architecture, they are evaluated with the ground truth depth maps generated from Pix4d. Thus, we have not evaluated the network performance for pose estimation or modified the pose network and has adapted the available, simple architecture that suits our objective as proposed by Godard et al., (2019). The dataset used to train our model and the results from our network architecture are evaluated in the following section.

4. Experiments

In this section, we introduce the dataset and then describe details of our implementation. After that, we present a quantitative and qualitative comparison. Furthermore, we conduct several ablation studies.

4.1. Dataset

The UAV oblique video dataset used for our work is taken from the UAVid (Lyu et al., 2020) imageries. The UAVid dataset is a combination of 42 video sequences captured from Wuhan (China) and Gronau (Germany). The original frame rate is 20 frames/second and the images are captured at a flying height of 50 ~ 100 m with a flying speed of 10 m/s, at an angle of 45°. The images are of size 4096 × 2160 and 3840 × 2160 for Germany and China respectively. There are 34 video sequences for China and 9 video sequences for Germany. The images from China are captured at varying height and have lots of dynamic features like cars, pedestrians, crowded streets which add extra complexity on top of the obliqueness of the view. Also, the images from China have a higher depth range, reaching to more than 500 m. The training dataset from Germany in contrast, covers a smaller region, has a depth range of 150 m, consists of rooftops, vegetation and is more uniform. The number of images used for training, testing and evaluation from each dataset are specified in Table 1. Some training sample images from both datasets are shown in Fig. 5. The used training dataset consists of three categories, images from Germany alone, images from China alone and both combined together.

The frame rate is an important parameter to train the model. The original frame rate of 20 frames/second, produced various noises in the results possibly due to the smaller base and parallax error. Several trials are carried out for both Germany and China dataset to determine the optimal frame rate. For Germany dataset, this is found to be 10 frames/second, while for China dataset the selection required further reduction in overlapped images. As discussed above due to a large number of high-rise buildings in Chinese dataset, the flying height had to be controlled manually. This led to varying heights and selection of a fixed frame rate due to different overlaps is complex. After several trials, it is found that there are significant noises for all frame rates due to the smaller base between many consecutive frames. In order to overcome this, frames with very narrow based are removed manually from the video sequence and then a frame rate of 1 frame/second is used for learning from the dataset.

Table 1
Dataset specification.

	Germany (4096 × 2160)	China (3840 × 2160)
Training	11,438	25,800
Validation	1500	2400
Testing	409	88

To test the model performance, the depth maps generated by the model is compared with point clouds generated using Pix4D which is considered as reference depths. The 3D reconstruction in Pix4D also encountered the problem of a narrow base between successive frames in the datasets. To widen the base and to create good quality point clouds some of the frames had to be removed which decreased the number of available reference frames. The reference depth images are theme derived from the point cloud using the P-matrix obtained from Pix4D. A total of 409 images for Germany and 88 images for China is obtained as reference depths through this process. For testing the model, a single test image is given as input to the model from which a depth map is produced which is then compared to the corresponding reference depth.

4.2. Reference depths

To assess the quality of the generated single depth images from various models, the results from these models are compared with ground truth depth images. For our work, the ground truth images are prepared from point clouds obtained from a commonly used Photogrammetric tool (Pix4D). For Germany dataset, GCP points are collected to serve as control points while generating the point clouds. This improved the quality of the generated point clouds for Germany dataset. The point clouds obtained from Pix4D are in real world coordinate system (WGS 1984, UTM 32 N) with mean sea level as a datum. These point clouds are interpolated to form reference depth images that are compared with the test images from Germany and China dataset. To validate the model performance, the obtained inverted depths are converted to depths using an appropriate scale based on the reference depth images.

4.3. Implementation details

All models in this work are implemented in PyTorch (Paszke et al., 2017) with an input resolution of 640 × 352 pixels for both datasets. Our model is trained for 40 epochs, 20 epochs, 20 epochs each for Germany, China dataset and both datasets mixed together. We use Adam optimizer for training all three datasets. The learning rate is set to 10^{-4} for 75% of the epochs and 10^{-5} for remaining epochs. Training takes around 17, 22 and 27 h in a single Nvidia Titan Xp GPU of 16 GB memory for Germany, China and combined dataset respectively. After hyperparameter tuning, the batch size is fixed at 12. The different weights for the loss terms are tuned and the respective weights of $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are determined after several experiments. The weightage of reprojection loss, auto masking loss are chosen as 1, while the weightage of smoothness loss is maintained as 0.001 and weight of contrastive loss is fixed as 0.5. The obtained disparities are converted to depths using an appropriate scale based on the model and dataset tested. In order to convert our disparities, we follow a similar scaling method as implemented by other SMDE models like Zhou et al., (2017), Repala and Dubey, (2018), Aleotti et al., (2018), Godard et al., (2019) etc. Here, the minimum and maximum depths are obtained from the reference depths which are used to fix the depth range of the model results. The obtained disparity values are then multiplied with a scaling value that produces metrically comparable results with the reference depths. From a photogrammetric perspective, the orientation from the pose networks could also be used to find the appropriate scaling for the entire image. The pose network takes an average of all the training images to determine the relative orientation parameters. However due to the added complexities of this cannot be used in direct orientation of images and to make our results comparable with other SMDE models we follow the simpler method for scaling.

4.4. Evaluation metrics

To assess the performance, various pixel-wise metrics are calculated between the models and reference depths. The evaluation of the accuracy is done based on calculating several metrics between the single

Fig. 5. Training Video sequences from UAVid dataset (Lyu et al., 2020): (a) Germany (First row) and (b) China (second row).

image depths (d') generated from the model after fixing the model parameters and the reference depths (d) produced from PIX4D. These metrics are chosen based on several literatures related to monocular depth estimation and the use of similar metrics can help in understanding our model performance with the current dataset with other similar models. We followed the quantitative comparison with the specified metrics from authors Hermann et al., (2020), Godard et al., (2019), Zhou et al., (2017) to ease the evaluation. This includes Absolute Relative difference (Abs Rel) to calculate the average difference between the corresponding pixels from reference depths and model predicted depths for all images relative to reference depths as given in equation (8), Squared Relative difference (Sq Rel) which is the squared difference of absolute values between reference and model predicted depths relative to reference depths as given in equation (9), Root Mean Square Error (RMSE) given in equation (10), accuracy given in Eq. (11) as described in Godard et al., (2019) and Hermann et al., (2020).

$$\text{AbsRel} = \frac{1}{N}\sum_{i=1}^{N}\frac{|d(x_i) - d^{'}(x_i)|}{d(x_i)} \tag{8}$$

$$\text{SqRel} = \frac{1}{N}\sum_{i=1}^{N}\frac{|d(x_i) - d^{'}(x_i)|^2}{d(x_i)} \tag{9}$$

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d(x_i) - d'(x_i))^2} \tag{10}$$

$$\text{Accuracy}(\delta_\theta) = \frac{1}{N}\sum_{i=1}^{N} max(\frac{d(x_i)}{d^{'}(x_i)}, \frac{d^{'}(x_i)}{d(x_i)}) < \theta \tag{11}$$

Where $d(x_i)$ refers to the reference depth at each pixel (x) at i^{th} position, $d^{'}(x_i)$ refers to the model produced single image depths at each pixel at i^{th} position. Here Accuracy is measured as the fraction of pixels that are within a certain threshold θ to the corresponding pixel wise value in the reference depth map. The thresholds chosen are 25%, 15% and 5% based on the standard benchmarks from KITTI quantitative assessments (Garg et al., 2016).

4.5. Comparison of results

In this part, we evaluate our model results by comparing with reference depths generated from PIX4D software. Our model results are also compared with four other models, Hermann et al., (2020), Godard et al., (2019), Bian et al., (2019) and Zhou et al., (2017), as they have presented a self-supervised model for aerial image depth estimation with a similar pose network and some of the loss terms that we used. We have selected these models such that they can be compared with our approach. Hermann et al., (2020) used a self-supervised approach for estimating depth from single UAV images, Godard et al., (2019) used the state of the art network for self-supervised monocular depth estimation with terrestrial images and our architecture started by modifying their work, Bian et al., (2019), used a self-supervised approach for estimating the depth from monocular terrestrial images by including a geometric consistency loss term to reduce the scale issues using depth estimated from consecutive loss terms and Zhou et al., (2017) work has been the base for many of the current video based monocular depth estimation. All four models are trained with the same environment, dataset, resolution, batch size and number of epochs in order to make the results comparable. The computational efficiency of Nvidia GPU is around 10 images/ second with a pixel size of 640 × 352. The testing speed for Germany and China dataset of size 640x352 is same with the single Nvidia GPU. The computational times for training and testing for both datasets are given in Table 2. Though our model takes more time, our model helps in better reconstruction of certain objects in Germany dataset from the use of consecutive temporal frames.

4.6. Qualitative results

The results from various models along with reference depths are shown in Figs. 6a and 6b. From the qualitative aspect, our model results over Germany showed a closer approximation to that of the reference depths. The model reconstructs depth images that smoothen out the variations over vegetation and show fine edges of buildings and roofs. It shows that our model preserves small details. The objects closer to the camera are shown as yellow and farther regions are shown in blue. The local variations in the ground surface are difficult to differentiate. Similarly, for China dataset the smooth transition from closer to farther

Table 2
Computational times (Training & Testing).

Method	Training time (hrs)		Testing time (images per second)
	Germany	China	Germany & China
Hermann et al., (2020)	21	28	2.3
Godard et al., (2019)	19	29	2.7
Bian et al., (2019)	20	27	2.1
Zhou et al., (2017)	21	27	2.9
Our model	22	32	2.1

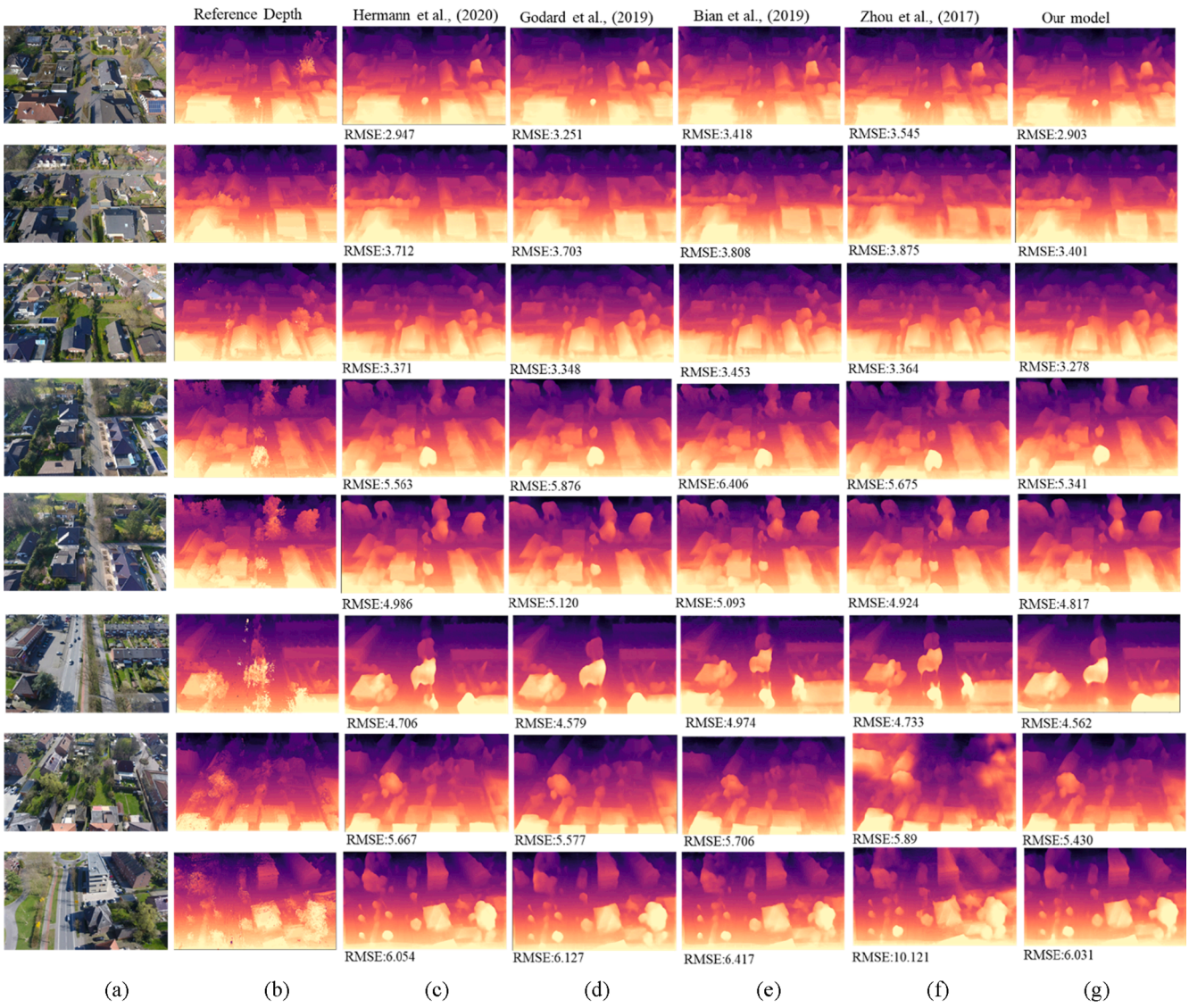

Fig. 6a. Qualitative comparison between (b) Reference depths from Pix4d, (c) Hermann et al., (2020), (d) Godard et al., (2019), (e) Bian et al., (2019), (f) Zhou et al., (2017), (g) Our model from Germany and China dataset. The test image is given in (a).

objects is visible. But in China dataset, due to larger depth variations and reduction in image resolution, each pixel contains many objects along with the dynamic objects making this dataset more challenging to predict. From images, it is difficult to differentiate the regions which have improved as all the models are well capable of generating depths images from UAV sequences. So to accurately understand the improvement in each model quantitative evaluation is performed.

4.7. Quantitative results

The quantitative metrics between various models, datasets and reference depths are shown in Table 3, Table 4 and Table 5. The generalising capability of the models trained over one dataset and tested on other datasets is estimated and the corresponding error metrics are given in Table 6. The values given in the table are expressed in terms of meters, as the scaling used to convert the disparities to depths are approximated from the reference depths expressed in meters.

From the tables, we could observe that the results vary between different datasets and models. The deep learning models are sensitive to training data and that is why various studies group the datasets based on the similarity of features like rural, urban and synthetic datasets and evaluate the model performance at different regions. We wanted to evaluate how well our model can generalise over different datasets that have complex features, dynamic objects and large depth variations. This makes Germany dataset a simpler dataset to learn from due to its simplicity and lack of moving objects compared to China dataset. The metrics for Germany dataset shows that our model is well capable of generating depths from single images that are closer to reference depths. The mean absolute difference between reference depths and model results are less compared to the other two models and also the percentage of pixels that are similar are more than 98% at a threshold of 25%. Also, from Table 3, we could see our model performs better in absolute terms compared to other models.

In higher depth ranges, like in China dataset which is used for training and testing, we could see from Table 4 that the RMSE values are larger than that of Germany dataset. For regions captured in China, farther points in oblique images might introduce errors, possibly due to reduction in image resolution resulting in many objects like trees, road, cars appear in just a single pixel. This along with the dynamic objects, moving cars and pedestrians, makes the process of learning highly challenging. Also, the maximum depth range in Germany dataset would be 150 m while for China it is more than 500 m. The inverse depths or the disparities generated from China datasets has pixel values less than 0.1. This means the pixel shift between images is less than 0.1 making it

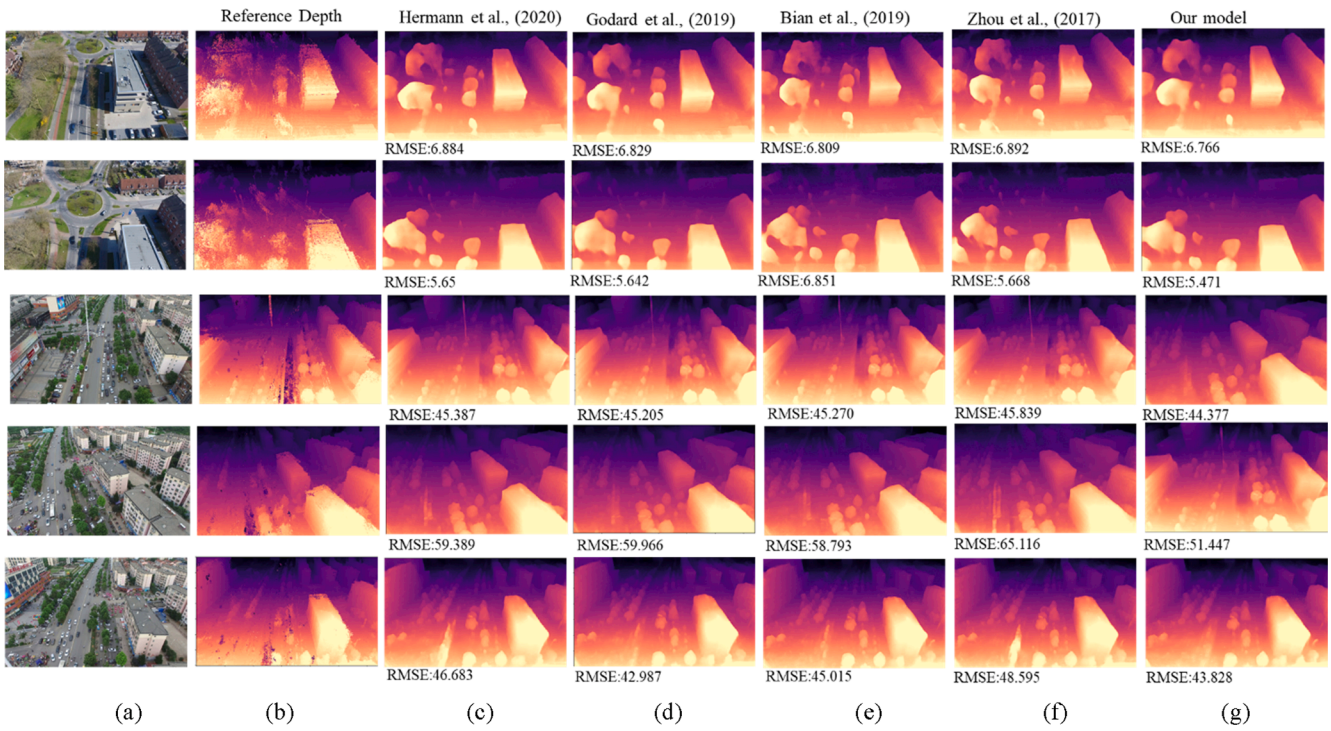

Fig. 6b. Qualitative comparison between (b) Reference depths from Pix4d, (c) Hermann et al., (2020), (d) Godard et al., (2019), (e) Bian et al., (2019), (f) Zhou et al., (2017), (g) Our model from Germany and China dataset. The test image is given in (a).

Table 3
Quantitative results achieved over different models with Germany dataset. The values represent the mean score over all the images in the corresponding test dataset.

Method	Training dataset	Testing dataset	Abs Rel	Sq Rel	RMSE	$\delta_{1.25}$ (Higher is better)	$\delta_{1.15}$ (Higher is better)	$\delta_{1.05}$ (Higher is better)
Hermann et al., (2020)	Germany	Germany	0.0372	0.302	4.275	0.9814	0.957	0.8085
Godard et al., (2019)	Germany	Germany	0.0371	0.295	4.1905	0.982	0.9582	0.810
Bian et al., (2019)	Germany	Germany	0.045	0.353	4.81	0.974	0.936	0.702
Zhou et al., (2017)	Germany	Germany	0.0573	0.869	5.971	0.923	0.894	0.747
Our model	Germany	Germany	**0.0366**	**0.2925**	**4.1735**	**0.982**	**0.9582**	**0.810**

Table 4
Quantitative results achieved over different models with China dataset. The values represent the mean score over all the images in the corresponding test dataset.

Method	Training dataset	Testing dataset	Abs Rel	Sq Rel	RMSE	$\delta_{1.25}$ (Higher is better)	$\delta_{1.15}$ (Higher is better)	$\delta_{1.05}$ (Higher is better)
Hermann et al., (2020)	China	China	0.109	7.858	49.828	0.874	0.755	0.337
Godard et al., (2019)	China	China	**0.105**	7.768	48.864	**0.879**	**0.764**	**0.376**
Bian et al., (2019)	China	China	0.106	7.750	48.541	0.875	0.763	0.373
Zhou et al., (2017)	China	China	0.113	9.920	55.524	0.869	0.754	0.314
Our model	China	China	0.109	**7.742**	**48.303**	0.878	0.761	0.327

Table 5
Quantitative results achieved over different models with both combined datasets. The values represent the mean score over all the images in the corresponding test dataset.

Method	Training dataset	Testing dataset	Abs Rel	Sq Rel	RMSE	$\delta_{1.25}$ (Higher is better)	$\delta_{1.15}$ (Higher is better)	$\delta_{1.05}$ (Higher is better)
Hermann et al., (2020)	Germany + China	Germany	0.0391	0.3181	4.42	0.981	0.956	0.786
Godard et al., (2019)	Germany + China	Germany	0.0392	0.3147	4.3722	0.981	0.955	0.786
Bian et al., (2019)	Germany + China	Germany	0.043	0.383	4.594	0.976	0.942	0.755
Zhou et al., (2017)	Germany + China	Germany	0.0485	0.853	5.646	0.965	0.936	0.757
Our model	Germany + China	Germany	**0.0381**	**0.322**	**4.364**	**0.982**	**0.956**	**0.797**
Hermann et al., (2020)	Germany + China	China	0.133	13.390	63.730	0.836	0.673	0.215
Godard et al., (2019)	Germany + China	China	0.132	11.840	60.631	0.839	0.670	0.225
Bian et al., (2019)	Germany + China	China	0.120	9.786	55.879	0.847	0.722	0.269
Zhou et al., (2017)	Germany + China	China	0.140	20.256	69.923	0.833	0.651	0.193
Our model	Germany + China	China	**0.117**	**9.664**	**54.855**	**0.859**	**0.734**	**0.302**

Table 6
Quantitative results achieved over different models trained over one dataset and tested on other to find the transferability potential of the model. The values represent the mean score over all the images in the corresponding test dataset.

Method	Training dataset	Testing dataset	Abs Rel	Sq Rel	RMSE	$\delta_{1.25}$ (Higher is better)	$\delta_{1.15}$ (Higher is better)	$\delta_{1.05}$ (Higher is better)
Hermann et al., (2020)	Germany	China	0.217	12.532	56.049	0.631	0.370	0.116
Godard et al., (2019)	Germany	China	0.215	**12.377**	55.832	0.636	0.376	0.117
Bian et al., (2019)	Germany	China	0.239	15.083	62.542	0.559	0.329	0.099
Zhou et al., (2017)	Germany	China	0.216	12.435	56.907	0.627	0.372	0.121
Our model	Germany	China	**0.214**	12.417	**55.048**	**0.645**	**0.379**	**0.118**
Hermann et al., (2020)	China	Germany	0.2054	4.231	17.3812	0.5076	0.2894	0.0936
Godard et al., (2019)	China	Germany	0.2016	3.766	16.17	0.523	0.2994	0.0962
Bian et al., (2019)	China	Germany	0.199	3.714	14.621	0.533	0.311	0.105
Zhou et al., (2017)	China	Germany	0.213	4.139	17.095	0.0505	0.2937	0.097
Our model	China	Germany	**0.19**	**3.0494**	**14.4122**	**0.5548**	**0.3298**	**0.11**

difficult for estimating depth. Increasing the base between images did not reduce this problem as it introduced errors due to occlusions between objects in image reconstruction task due to the larger shift. Several experiments have been carried out to find an appropriate base that reduces the impact of occlusions while having sufficient base for 3D reconstruction. Even after these experiments, there is the presence of pixels with very less disparity values (lesser than 0.1), providing more uncertainty for farther objects. To further evaluate this, instead of using the entire image, we used only pixels with depths less than 200 m for evaluation of error metrics. The obtained RMSE values are reduced by more than half the current numbers presented in Table 4. Whereas all models are having difficulty in learning from the China dataset, in comparative terms our model still performs better than the other models. It achieved an accuracy of almost 88% at a threshold level of 25%.

Finally, we combined images from both Germany and China dataset to understand how well the model performs when it has a diverse range of images during training. To evaluate the performance, the obtained model was tested on Germany and China dataset separately. From Table 5, we could see that the model performs comparatively better with Germany dataset than China dataset. Also, we can see that training with individual datasets produces better results than combining them. Here also we could observe the pattern of our model performing better than the other two models. The difference between our model with other models is higher in China dataset compared to Germany dataset. This could also show that as the depth variations increase and the difficulty in handling farther objects are handled slightly better by our model in comparison to the other two.

In order to test the model transferability and its ability to generalise, we trained the model with images from one dataset and tested it on the other. From Table 6, we can see that the model performance is reduced, still all models are capable of generating depth maps which are trained in different depth ranges. This is mainly due to the exposure of model with one particular landscape which caused the reduction in various metrics. Our model requires only images for training due to its self-supervised nature, so training in a different region can be handled through training with new image sequences for improvement or our model can be fine-tuned on other image sequence based on the requirement.

4.8. Ablation study

To study the impact of various loss terms and the image resolution, we carried out various experiments to understand the model performance. For quantitative evaluation, we used only Germany dataset for these experiments as they showed closer approximation to the ground truth than other datasets.

Scratch: Our model uses pre-trained encoders for the training process. In order to understand the effect of pre-trained models, we experimented by training our encoders from scratch. From Table 7, we could see that pre-training improves the results and there is a reduction in an absolute difference between reference depths and our model result. Also, training a model with pre-trained weights might help the model in finding the global minima easier compared to training from scratch.

Input resolution: As we could see from Table 7, input resolution has a large impact on the results due to mixed features within a pixel. Due to the limitation of the graphics card, we opted 640 × 352 resolution for our model. UAV images are of high resolution and reducing the images to lower size introduced uncertainties in results. To further see this, we also reduced the image resolution to half the resolution from our model size (320X192 pixels) and trained the model. The reduction in model generation capability and the errors caused due to the smoothening of pixels are seen in Fig. 7.

Loss terms: To further assess the effects of loss terms on our model, we tested by removing and adding different loss terms and different combinations as shown in Fig. 8. As we could see from Table 7, our proposed model with all four loss terms performed better than all the other combinations. It is understood that each loss contributes a little to model improvement. Adding contrastive loss terms without masking loss also shows closer performance to our proposed model. The use of contrastive loss for improving the image generation is clearly visible from these metrics. The RMSE values between reference depth and model generated results shows a reduction in values when the contrastive loss is included. In Table 7, we could observe that the absolute values for all metrics between models without contrastive loss and after using contrastive loss showed an improvement.

Sensitivity Analysis: To understand the variability in model performance, we ran the model with five subsets of Germany dataset. For each run, the subset will include at least 70% of the total images and for all the subsets the model is run with same environmental parameters. The evaluation metrics for each subset is shown in Table 8. The mean and standard deviation from these repeated experiments are also specified.

The results from Table 8, indicates that our model performs consistently well over the five different subsets from the Germany dataset. The mean value is 4.166 from the five subsets which is closer to our model result with whole dataset as 4.173. This relation can also be seen from the standard deviation of 0.0091, showing the minimal deviation from average value. Also, the correlation between RMSE from two different subsets are more than 0.999 for all cases.

5. Conclusion

This paper presents a novel approach for estimating depth information from an oblique UAV video. Our model is based on a self-supervised approach that does not require ground truth depths for training and can be trained with a UAV video sequence captured over different regions. Our model learns both depth and pose information using two networks

Table 7
Quantitative results with different experiments to our model trained and tested over Germany dataset. The values represent the mean score over all the images in the corresponding test dataset.

Method	Abs Rel	Sq Rel	RMSE	$\delta_{1.25}$ (Higher is better)	$\delta_{1.15}$ (Higher is better)	$\delta_{1.05}$ (Higher is better)
Baseline- pretrained with four loss (640 × 352)	**0.0366**	**0.2925**	**4.1735**	**0.982**	**0.9582**	0.810
Baseline- Scratch with four loss	0.0391	0.3091	4.287	0.9821	0.956	0.7867
Baseline- pretrained with four loss smaller resolution (320 × 192)	0.0481	0.3606	4.771	0.982	0.948	0.662
Baseline- pretrained with three loss (reprojection, smoothness and contrastive loss)	0.0366	0.294	4.195	0.982	0.958	**0.811**
Baseline- pretrained with three loss (reprojection, smoothness and Masking loss)	0.0372	0.3014	4.234	0.981	0.957	0.8087
Baseline- pretrained with two loss (reprojection and smoothness loss)	0.0366	0.294	4.201	0.982	0.957	0.809

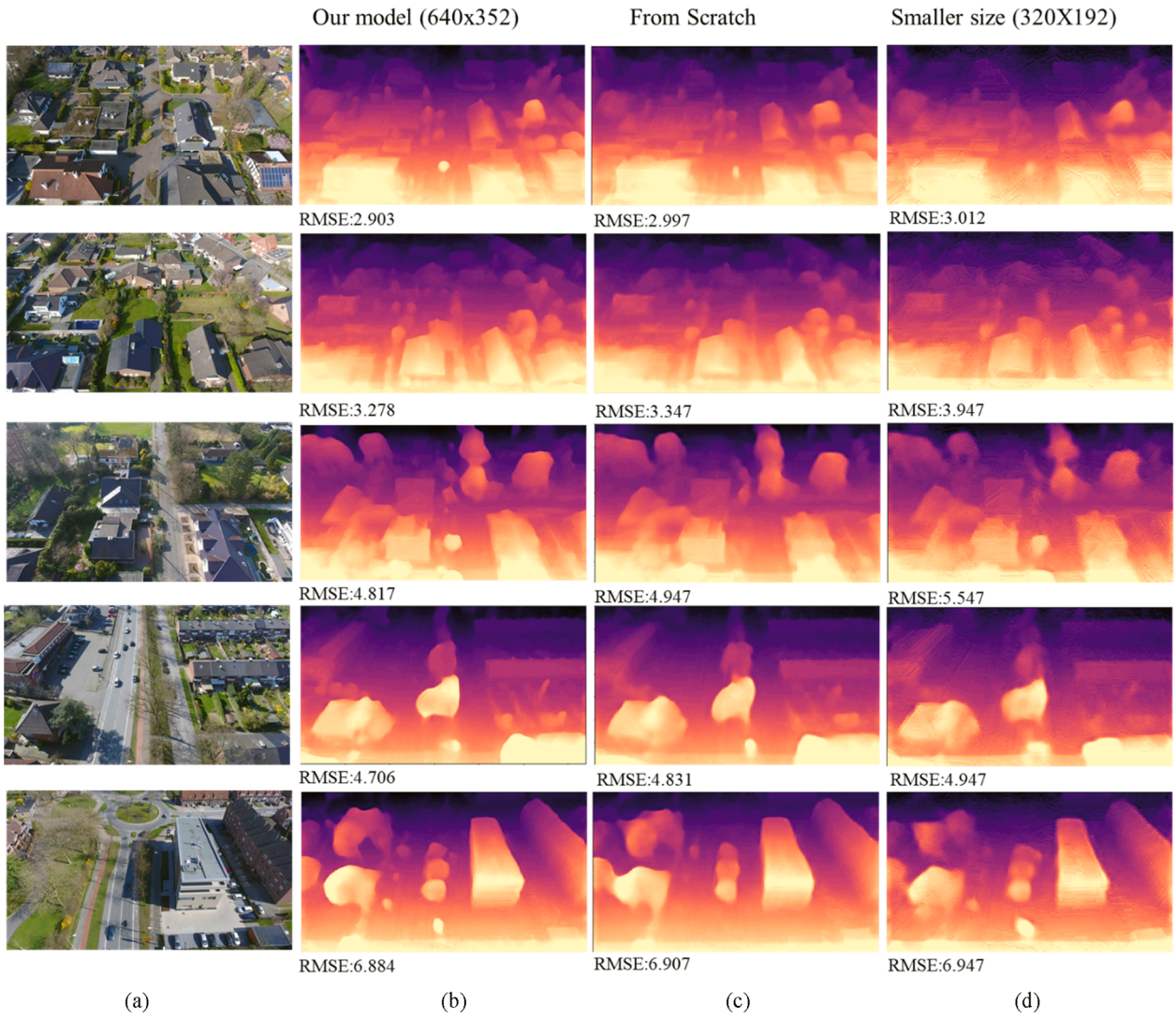

Fig. 7. Qualitative results: (b) Our model, (c) Model trained from scratch, (d) Model trained with smaller resolution. The test image is given in (a).

during the training stage. The depth network consists of two encoders to capture as much image information from consecutive images and pose network calculates the relative camera position from three images. Only depth network is used for the inference stage to predict the depths from single UAV images reducing the processing time significantly. The contrastive loss term is added to improve the image generation and to increase the similarity between reconstructed images. Our model performs the best over Germany subset in UAVid video dataset due to its smaller depth range and less complex scene. The predicted depths from our model compare well with the reference depths and are also better than other state-of-the-art methods. The model performance over regions with dynamic objects and larger depth variations are few short comings of our model. For future work, we would like to explore further the depth estimation model for very large depth ranges in a scene along with the dynamic objects.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence

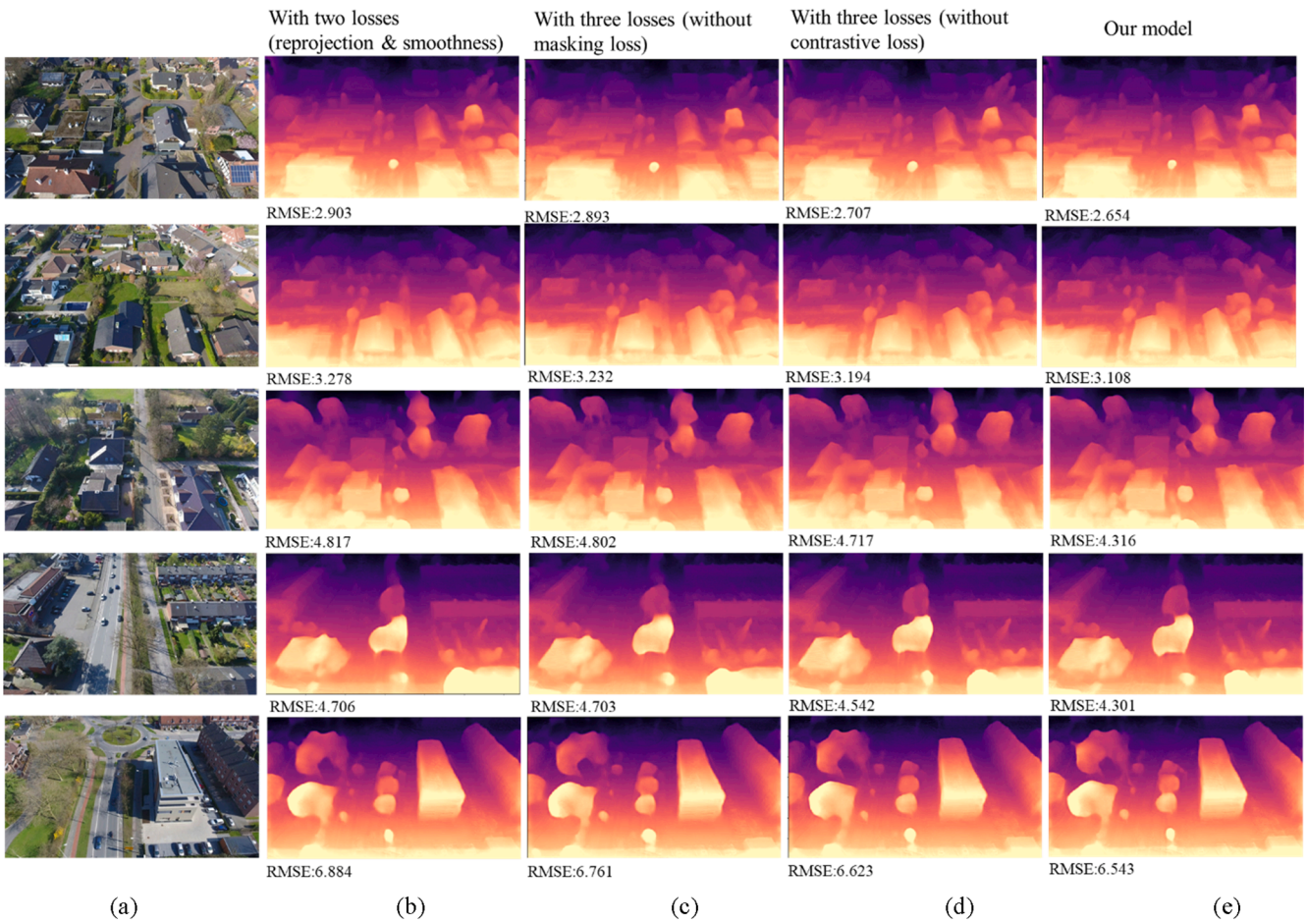

Fig. 8. Qualitative results: (b) Model trained with two losses (reprojection & smoothness loss), (c) Model trained with three losses (without masking loss), (Godard et al., 2019) (d) Model trained with three losses (without contrastive loss), (e) Our model. The test image is given in (a).

Table 8
Quantitative results achieved over five different subsets from Germany dataset with our model. The values represent the mean score over all the images in the corresponding test dataset.

Method	Testing dataset	Abs Rel	Sq Rel	RMSE	$\delta_{1.25}$ (Higher is better)	$\delta_{1.15}$ (Higher is better)	$\delta_{1.05}$ (Higher is better)
Our model -Germany subset1	Germany	0.0372	0.297	4.156	0.981	0.956	0.808
Our model -Germany subset2	Germany	0.0367	0.291	4.163	0.982	0.958	0.810
Our model -Germany subset3	Germany	0.0367	0.292	4.157	0.982	0.958	0.814
Our model -Germany subset4	Germany	0.0367	0.295	4.175	0.982	0.958	0.815
Our model -Germany subset5	Germany	0.0367	0.292	4.174	0.982	0.958	0.806
Mean	Germany	0.0368	0.294	4.166	0.982	0.957	0.811
Standard deviation	Germany	0.000227	0.00222	0.0091	0.000360	0.000671	0.00322

the work reported in this paper.

References

Aicardi, I., Chiabrando, F., Grasso, N., Lingua, A.M., Noardo, F., Spanó, A., 2016. UAV photogrammetry with oblique images: First analysis on data acquisition and processing. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 835–842 https://doi.org/10.5194/isprsarchives-XLI-B1-835-2016.

Alagoz, B.B., 2016. A Note on Depth Estimation from Stereo Imaging Systems. Anatol. Sci. 1, 8–13.

Aleotti, F., Tosi, F., Poggi, M., Mattoccia, S., 2018. Generative Adversarial Networks for Unsupervised Monocular Depth Prediction. ECCV. 337–354. https://doi.org/10.1007/978-3-030-11009-3_20.

Amirkolaee, H.A., Arefi, H., 2019. Height estimation from single aerial images using a deep convolutional encoder-decoder network. ISPRS J. Photogramm. Remote Sens. 149, 50–66. https://doi.org/10.1016/j.isprsjprs.2019.01.013.

Bhandare, A., Bhide, M., Gokhale, P., Chandavarkar, R., 2016. Applications of Convolutional Neural Networks. Int. J. Comput. Sci. Inf. Technol. 7, 2206–2215.

Bian, J., Li, Z., Wang, N., Zhan, H., Shen, C., Cheng, M., Reid, I., 2019. Unsupervised Scale-consistent Depth and Ego-motion Learning from Monocular Video. In: Thirty-Third Conference on Neural Information Processing Systems, pp. 1–11.

Chen, K., Weinmann, M., Sun, X., Yan, M., Hinz, S., Jutzi, B., Weinmann, M., 2018. SEMANTIC SEGMENTATION of AERIAL IMAGERY VIA MULTI-SCALE SHUFFLING CONVOLUTIONAL NEURAL NETWORKS with DEEP SUPERVISION. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 4, 29–36. https://doi.org/10.5194/isprs-annals-IV-1-29-2018.

Dai, Q., Patil, V., Hecker, S., Dai, D., Van Gool, L., Schindler, K., 2019. Self-supervised Object Motion and Depth Estimation from Video, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

Eigen, D., Puhrsch, C., Fergus, R., 2014. Depth Map Prediction from a Single Image using a Multi-Scale Deep Network. NIPS 14, 1–9.

Furukawa, Y., Hernández, C., 2013. Multi-view stereo: A tutorial. Found. Trends Comput. Graphics Vision 9 (1-2), 1–148. https://doi.org/10.1561/0600000052.

Garg, R., Vijay Kumar, B.G., Carneiro, G., Reid, I., 2016. Unsupervised CNN for single view depth estimation: Geometry to the rescue, in: European Conference on Computer Vision (ECCV 2016). pp. 740–756. https://doi.org/10.1007/978-3-319-46484-8_45.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? the KITTI vision benchmark suite, in. In: Proceedings of the IEEE Computer Society Conference

on Computer Vision and Pattern Recognition. https://doi.org/10.1109/CVPR.2012.6248074.
Godard, C., Aodha, O. Mac, Firman, M., Brostow, G., 2019. Digging into self-supervised monocular depth estimation. Proc. IEEE Int. Conf. Comput. Vis. 2019-Octob, 3827–3837. https://doi.org/10.1109/ICCV.2019.00393.
Godard, C., Mac Aodha, O., Brostow, G.J., 2017. Unsupervised monocular depth estimation with left-right consistency, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. pp. 6602–6611. https://doi.org/10.1109/CVPR.2017.699.
Guizilini, V., Ambrus, R., Pillai, S., Raventos, A., Gaidon, A., 2020. 3D Packing for Self-Supervised Monocular Depth Estimation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2016-Decem, 770–778. https://doi.org/10.1109/CVPR.2016.90.
Hermann, M., Ruf, B., Weinmann, M., Hinz, S., 2020. Self-Supervised Learning for Monocular Depth Estimation From Aerial Imagery. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. V-2–2020, 357–364. https://doi.org/10.5194/isprs-annals-v-2-2020-357-2020.
Hirschmuller, H., 2008. Stereo processing by semiglobal matching and mutual information. IEEE Trans. Pattern Anal. Mach. Intell. 30 (2), 328–341. https://doi.org/10.1109/TPAMI.2007.1166.
Julian, K., Mern, J., Tompa, R., 2017. UAV Depth Perception from Visual Images using a Deep Convolutional Neural Network. Tech. ReP. 1–7.
Kanatani, K. ichi, Chou, T.C., 1989. Shape from texture: General principle. Artif. Intell. 38, 1–48. https://doi.org/10.1016/0004-3702(89)90066-0.
Kang, S.B., Webb, J.A., Zitnick, C.L., Kanade, T., 1999. Multibaseline stereo system with active illumination and real-time image acquisition. In: IEEE International Conference on Computer Vision, pp. 88–93 http://doi.org/10.1109/iccv.1995.466802.
Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A., 2017. End-to-end learning of geometry and context for deep stereo regression. IEEE International Conference on Computer Vision (ICCV).
Koch, T., Liebel, L., Fraundorfer, F., Körner, M., 2019. Evaluation of CNN-based single-image depth estimation methods. Comput. Vis. Image Underst. 11131 LNCS, 331–348. https://doi.org/10.1007/978-3-030-11015-4_25.
Laina, I., Rupprecht, C., Belagiannis, V., Tombari, F., Navab, N., 2016. Deeper depth prediction with fully convolutional residual networks, in: Proceedings - 2016 4th International Conference on 3D Vision, 3DV 2016. pp. 239–248. https://doi.org/10.1109/3DV.2016.32.
Li, J., Yuce, C., Klein, R., Yao, A., 2017. A two-streamed network for estimating fine-scaled depth maps from single RGB images. Comput. Vis. Image Underst. 186, 25–36. https://doi.org/10.1016/j.cviu.2019.06.002.
Li, Qing, Zhu, J., Liu, J., Cao, R., Li, Qingquan, Jia, S., Qiu, G., 2020. Deep Learning based Monocular Depth Prediction: Datasets. Methods and Applications, Arxiv.
Liang, Z., Feng, Y., Guo, Y., Liu, H., Chen, W., Qiao, L., Zhou, L., Zhang, J., 2018. Learning for disparity estimation through feature constancy, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Liu, F., Shen, C., Lin, G., 2015. Deep convolutional neural fields for depth estimation from a single image, in. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 5162–5170. https://doi.org/10.1109/CVPR.2015.7299152.
Lyu, Y., Vosselman, G., Xia, G.S., Yilmaz, A., Yang, M.Y., 2020. UAVid: A semantic segmentation dataset for UAV imagery. ISPRS J. Photogramm. Remote Sens. 165, 108–119. https://doi.org/10.1016/j.isprsjprs.2020.05.009.
Madhuanand, L., Nex, F., Yang, M.Y., 2020. Deep Learning for Monocular depth Estimation From UAV Images. ISPRS Ann. Photogramm., Remote Sens. Spat. Inform. Sci. V-2-2020, 451–458.
Mahjourian, R., Wicke, M., Brain, G., 2018. Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 5667–5675.
Mayer, N., Ilg, E., Hausser, P., Fischer, P., 2016. A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. CVPR. https://doi.org/10.1021/jo00301a029.
Mehta, I., Sakurikar, P., Narayanan, P.J., 2018. Structured adversarial training for unsupervised monocular depth estimation. Proc. - 2018 Int. Conf. 3D Vision, 3DV 2018 314–323. https://doi.org/10.1109/3DV.2018.00044.
Mou, L., Zhu, X.X., 2018. IM2HEIGHT: Height Estimation from Single Monocular Imagery via Fully Residual Convolutional-Deconvolutional Network. Arxiv. 1–13.
Nex, F., Gerke, M., Remondino, F., Przybilla, H.J., Bäumker, M., Zurhorst, A., 2015. Isprs benchmark for multi-platform photogrammetry. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 2, 135–142. https://doi.org/10.5194/isprsannals-II-3-W4-135-2015.
Nex, Francesco, Remondino, Fabio, 2014. UAV for 3D mapping applications: A review. Appl. Geomatics 6 (1), 1–15. https://doi.org/10.1007/s12518-013-0120-x.
Paszke, A., Chanan, G., Lin, Z., Gross, S., Yang, E., Antiga, L., Devito, Z., 2017. Automatic differentiation in PyTorch 1–4.
Pix4D (version 4.4.12) [WWW Document], 2020. URL https://www.pix4d.com/ (accessed 4.18.20).
Poggi, M., Aleotti, F., Tosi, F., Mattoccia, S., 2018. Towards Real-Time Unsupervised Monocular Depth Estimation on CPU. IEEE Int. Conf. Intell. Robot. Syst. 5848–5854 https://doi.org/10.1109/IROS.2018.8593814.
Remondino, F., Spera, M.G., Nocerino, E., Menna, F., Nex, F., Gonizzi-Barsanti, S., 2013. Dense image matching: Comparisons and analyses. Proc. Digital Heritage 2013, 47–54. https://doi.org/10.1109/DigitalHeritage.2013.6743712.
Repala, V.K., Dubey, S.R., 2018. Dual CNN Models for Unsupervised Monocular Depth Estimation, in: Pattern Recognition and Machine Intelligence. p. 9.
Saxena, A., Chung, S.H., Ng, A.Y., 2005. Learning depth from single monocular images. Adv. Neural Inf. Process. Syst. 1161–1168.
Spencer, J., Bowden, R., Hadfield, S., 2020. DeFeat-Net: General Monocular Depth via Simultaneous Unsupervised Representation Learning. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 14402–14413.
Szeliski, R., Zabih, R., 2000. An experimental comparison of stereo algorithms, Triggs B., Zisserman A., Szeliski R. (eds) Vision Algorithms: Theory and Practice. IWVA 1999. Lecture Notes in Computer Science, vol 1883. https://doi.org/10.1007/3-540-44480-7_1.
Tan, F., Zhu, H., Cui, Z., Zhu, S., Pollefeys, M., Tan, P., 2020. Self-Supervised Human Depth Estimation from Monocular Videos, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition.
Tosi, F., Aleotti, F., Poggi, M., Mattoccia, S., 2019. Learning monocular depth estimation infusing traditional stereo knowledge, in. IEEE Conference on Computer Vision and Pattern Recognition.
Vallet, J., Panissod, F., Strecha, C., Tracol, M., 2012. Photogrammetric Performance of an Ultra Light Weight Swinglet "UAV". ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVIII-1/, 253–258. https://doi.org/10.5194/isprsarchives-xxxviii-1-c22-253-2011.
van den Heuvel, Frank A., 1998. 3D reconstruction from a single image using geometric constraints. ISPRS J. Photogramm. Remote Sens. 53 (6), 354–368. https://doi.org/10.1016/S0924-2716(98)00019-7.
Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K., 2017. SfM-Net: Learning of Structure and Motion from Video, in: Arxiv.
Voumard, Jérémie, Derron, Marc-Henri, Jaboyedoff, Michel, Bornemann, Pierrick, Malet, Jean-Philippe, 2018. Pros and cons of structure for motion embarked on a vehicle to survey slopes along transportation lines using 3D georeferenced and coloured point clouds. Remote Sens. 10 (11), 1732. https://doi.org/10.3390/rs10111732.
Wang, C., Buenaposada, J.M., Zhu, R., Lucey, S., 2018. Learning Depth from Monocular Videos Using Direct Methods. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2022–2030. https://doi.org/10.1109/CVPR.2018.00216.
Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P., 2004. Image quality assessment: From error visibility to structural similarity. IEEE Trans. Image Process. 13 (4), 600–612. https://doi.org/10.1109/TIP.2003.819861.
Žbontar, J., Le Cun, Y., 2015. Computing the stereo matching cost with a convolutional neural network. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 07–12-June, 1592–1599. https://doi.org/10.1109/CVPR.2015.7298767.
Zhang, R., Tsai, P.S., Cryer, J.E., Shah, M., 1999. Shape from shading: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 21, 690–706. https://doi.org/10.1109/34.784284.
Zhou, Kun, Meng, Xiangxi, Cheng, Bo, 2020. Review of Stereo Matching Algorithms Based on Deep Learning. Comput. Intell. Neurosci. 2020, 1–12. https://doi.org/10.1155/2020/8562323.
Zhou, T., Brown, M., Snavely, N., Lowe, D.G., 2017. Unsupervised Learning of Depth and Ego-Motion from Monocular Video. IEEE Conf. Comput. Vis. Pattern Recognit. https://doi.org/10.1109/CVPR.2017.700.