12-2022

Depth Estimation Using 2D RGB Images

Taher Naderi
University of Tennessee, Knoxville, tnaderi@vols.utk.edu

To the Graduate Council:

I am submitting herewith a dissertation written by Taher Naderi entitled "Depth Estimation Using 2D RGB Images." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Hairong Qi, Major Professor

We have read this dissertation and recommend its acceptance:

Hairong Qi, Amir Sadovnik, Seddik M. Djouadi, Jason P. Hayward

Accepted for the Council:

Dixie L. Thompson

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Depth Estimation Using 2D RGB Images

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Taher Naderi

December 2022

To my mom, Manijeh Ranjbaran.

Acknowledgements

I would like to thank Dr. Qi, Dr. Sadovnik, Dr. Djouadi, and Dr. Hayward for accepting to be in my committee. Specifically, I would like to thank my advisors Dr. Qi and Dr. Sadovnik for their support during these years.

As the area of our knowledge grows, so too does the perimeter of our ignorance.

Neil deGrasse Tyson

Abstract

Single image depth estimation is an ill-posed problem. That is, it is not mathematically possible to uniquely estimate the 3rd dimension (or depth) from a single 2D image. Hence, additional constraints need to be incorporated in order to regulate the solution space. As a result, in the first part of this dissertation, the idea of constraining the model for more accurate depth estimation by taking advantage of the similarity between the RGB image and the corresponding depth map at the geometric edges of the 3D scene is explored.

Although deep learning based methods are very successful in computer vision and handle noise very well, they suffer from poor generalization when the test and train distributions are not close. While, the geometric methods do not have the generalization problem since they benefit from temporal information in an unsupervised manner. They are sensitive to noise, though. At the same time, explicitly modeling of a dynamic scenes as well as flexible objects in traditional computer vision methods is a big challenge.

Considering the advantages and disadvantages of each approach, a hybrid method, which benefits from both, is proposed here by extending traditional geometric models' abilities to handle flexible and dynamic objects in the scene. This is made possible by relaxing geometric computer vision rules from one motion model for some areas of the scene into one for every pixel in the scene. This enables the model to detect even small, flexible, floating debris in a dynamic scene. However, it makes the optimization under-constrained. To change the optimization from under-constrained to over-constrained while maintaining

the model's flexibility, "moving object detection loss" and "synchrony loss" are designed. The algorithm is trained in an unsupervised fashion.

The primary results are in no way comparable to the current state of the art. Because the training process is so slow, it is difficult to compare it to the current state of the art. Also, the algorithm lacks stability. In addition, the optical flow model is extremely noisy and naive. At the end, some solutions are suggested to address these issues.

Table of Contents

Chapter 1

Introduction

Depth estimation is an important step in understanding the geometry of a 3D scene. In addition, many downstream applications, such as 3D modeling, navigation in robotics, autonomous driving, 3D video stabilization [192], augmented reality (AR) and special video effects [299], and converting videos for virtual reality (VR) viewing [125], etc., rely on accurate depth estimation. Based on sensor design and methodologies related to the structure of sensors, depth estimation can be categorized as

- **Active sensor/method**: A sensor/method is called active if it sends a signal to the environment itself and gathers the information from the reflection of the environment.
- **Passive sensor/method**: The sensor or method is called passive if it uses the signal already available in the environments.

Active sensors/methodologies rely on sending stimulus to the environment to estimate the depth of the scene using that stimulus. They include Radar, LIDAR, RGBD cameras, and Ultrasound devices [74]. However, each of them has its own issues. For example, RGB-D cameras suffer from limited range of measurements, estimation of depth using LIDAR and Radar are sparse, and Ultrasound devices are inherently imprecise. In addition to the aforementioned issues, these are energy-consuming devices and they are large, which is an issue when one is thinking about small robots like micro aerial vehicles. Differently, RGB

cameras are less costly and light. Having this in mind, it would be valuable to search for depth estimation algorithms that depend on color images just like human beings, primates, and birds (of pray) which all benefit from advanced vision systems that help navigate through obstacles easily. This is called monocular depth estimation (MDE) which is the focus of this dissertation.

1.1 Problem definition

Let $I \in \mathbb{R}^{3\times h\times w}$ be a single 3-channel color image. Assume the spatial dimensions of the image is $w \times h$. Assume $D \in \mathbb{R}^{1\times h\times w}$ is the depth map of the image I. Then monocular depth estimation (MDE) is casted as finding a non-linear mapping Ψ such that

$$\begin{aligned} \Psi : \mathbb{R}^{3\times h\times w} &\longrightarrow \mathbb{R}^{1\times h\times w} \\ D &= \psi(I) \end{aligned} \tag{1.1}$$

This has been illustrated in Fig. 1.1. On the other hand, one might be interested to design an algorithm that extract depth map as a result the structure, i.e. depth map of the scene from a train of images. In this case the definition would extend into

$$\begin{aligned} \Psi : \mathbb{R}^{N\times 3\times h\times w} &\longrightarrow \mathbb{R}^{M\times 1\times h\times w},\ M \leq N \\ D_m = \psi(\{I_t | t = 1, 2, 3, \cdots, N\}),&\ m \in \{1, 2, 3, \cdots, N\} \end{aligned} \tag{1.2}$$

m is usually the central image in the sequence of $1, 2, \cdots, N$. This has been illustrated in Fig. 1.2. While the depth map estimation using MDE methods is less costly in computations and is not sensitive to degeneracy issue as well as calibration difficulties of stereo rigs, it is an ill-posed problem. The reason is simple. A 2D image can be generated from countless different distinct 3D scenes. Therefore, a MDE algorithm must benefit from different cues which are called **monocular cues**.

1.2 Monocular cues

Monocular cues are visual cues to understand depth of scene that can be perceived with only one eye. People who are visually impaired in one eye are still able to understand the 3D structure of the scene, although with less accuracy. So they rely on monocular cues for navigation. These cues include motion parallax, interposition or occlusion, and perspective. These cues are available in works of art (see Figs 1.6 and 1.5). The artists exploit these cues so that viewers feel like they are looking at a three-dimensional environment.

Motion parallax: When one is in a moving car, he/she feels like distant objects move more slowly and closer objects move faster when looking through the side window. This is because our brain is able to interpret this difference in lateral speed as depth.

Size and height: One important monocular cue is size and height of known objects. For example a small car is interpreted as further away while a larger sedan car is perceived closer to the viewer. This might be utilized in estimation of motion in depth direction to avoid collision with objects [342, 343].

Perspective: Linear perspective is the tendency of far away lines to seemingly converge. It is an important monocular cue for depth perception. The places where these lines converge are called vanishing points. The location of objects in the scene compared to those lines can be perceived. The seemingly converging tracks of train is an example. There are several types of linear perspective, including 1-point perspective, 2-point perspective, 3-point perspective, etc. Mathematically speaking though, all of them can be expressed with one formula. See Fig. 1.3 and 1.5.

Atmospheric or aerial perspective: Another type of monocular cue is texture gradient. The textures appear in the scene with more details when they are closer and distant objects are fuzzy, pale. This happens because of atmospheric interruptions, like dust, and provides

clues about the distance of objects in the scene. See Figs 1.4 and 1.6.

Interposition, or occlusion: Interposition, or occlusion is a monocular cue. Our brain estimates the depth of an object relative to another one if the first one covers the other one partially. Although we do it so naturally, it is a challenge for single image MDE [92] or even in dynamic scene [247].

1.3 Motivations

One might question the benefit of designing monocular depth estimation algorithms while stereo vision can avoid many challenges indigenous to MDE like scale ambiguity or problems like dynamic scene as well as difficulties in exploiting monocular cues. To answer this question, it should be noted that the stereo vision is limited to approximately 10 meters similar to D435 RGBD camera. This limitation springs from the sensitivity of the problem to the distance between stereo rigs. Farther than this distance, human vision mainly relies on monocular cues. So the first benefit is that the MDE has longer range than the stereo vision. On the other hand, having an efficient monocular depth estimation algorithm can benefit many single camera devices available almost everywhere. The third benefit of monocular video depth estimation is that such algorithms pave the way for an efficient exploitation of temporal information in stereo video depth estimation. Other benefits of using MDE in designing algorithms could be less computational cost and avoiding the degeneracy issues as well as calibration difficulties of stereo rigs.

At the end of this section It is worse mentioning that the monocular cues are usually defined by physics of the problem. Many single image MDEs just try to exploit the universality of CNNs with the hope that it would catch the monocular cues. However, it make sense that if one explicitly addresses the monocular cues with appropriate modules, he/she will get better results. It is not an easy task though.

1.4 Challenges and approaches

Broadly speaking, there are two different (passive) approaches available to estimate the depth of a scene. Traditional computer vision methods that rely on assumptions on camera models which result in pure geometric approaches, and deep learning based algorithms which consider a universal function, usually based on Deep Neural Networks (DNNs), and train it on an already recorded datasets, i.e. train dataset. These methods rely on similarity of distribution of data in the already recorded train datasets and the unknown test datasets. Deep learning based methods have remarkably enhanced the effectiveness of many computer vision problems including MDE. Also, the methods are usually rely on monocular cues and done in single image fashion.

However, single image depth estimation, as described before, is an ill-posed problem. On the other words, going from three-dimensional world to 2D images is a one-way function which is irreversible directly. During this process the information of the scene related to the third dimension is lost.

1.4.1 Single image MDE, challenges and methodologies

The minimal sensory setup for depth estimation is to use a single monocular image. However, recovering the scene's depth from a single image is an ill-posed problem that requires additional priors embedded in the model through learning-based methods to disambiguate different 3D interpretations. Existing deep learning methods can usually estimate accurate 2D depth maps. However, they lack local details and are often highly distorted when the maps are projected into 3D. This is due to the usage of down-sampling in the pretrained fully convolutional encoders, mostly designed for classification purpose. While feature resolution and granularity may not be important in performing tasks like image classification, they are crucial for dense prediction, where the architecture of the model should ideally be able to deliver features at or close to the resolution of the input image. Various techniques have been proposed to mitigate the above-mentioned issues.

One way is using dilated convolutions [350] to rapidly increase the receptive field without down-sampling. Another way is using skip connections from multiple stages of the encoder to the decoder [254]. By the same token, the problem has been addressed in [315] by connecting multi-resolution representations in parallel throughout the network. While all these techniques can solve the issue to some extent, they are subject to the problem of washed out information in deeper convolutional networks [124]. To mitigate the effect of these convolutions, some researchers have suggested to replace the building blocks entirely or in some places in networks by attention-based blocks [171, 194] or transformers [344, 246] which are themselves, based on attention mechanisms.

Even given that one can find a way to produce a high resolution depth map with many details by using skip connections, they might still run into an additional problem. One can explain this using the example shown in Fig. 1.7 that compares an RGB image and the corresponding depth map. The cabinet on the left and the table surface are almost texture-less in the RGB image and have gradient only at geometric edge locations in the depth map. On the other hand, the wall with the brick texture mainly shows a gradient-less area in the depth map but a lot of gradient in the RGB image. Looking at the high-pass filtered RGB image and the depth map suggests that most of the information needed to extract a depth map from a scene is near the geometric edges, i.e., edges in the RGB image which come from the geometric structure of the 3D scene. However, to extract the geometric edges, One needs to first remove the edges in the RGB image which mainly come from texture and color changes and replace the texture-less area of the RGB image with the deduced geometric structure in the depth map.

For this reason one might wish to give the convolutional neural network the ability to deduce the local geometric structure of the RGB image using guidance from the corresponding depth map. However, the depth map is not available at evaluation time. Instead, they can explore the idea of constraining the model by taking advantage of the similarity of the RGB features and the corresponding depth map features at geometric edges of the 3D scene for more accurate depth estimation. Hence a light-weight attention module

was proposed that uses the cross-correlation between the encoder and the decoder [224]. The functionality of this module can be interpreted as a guiding tool for an efficient feature extraction in the encoder and it can be used to merge the same size feature maps from the encoder to the decoder efficiently in any encoder-decoder structure with minimum added weight and computation burden to the base encoder-decoder network to address any task at hand.

The proposed module along with the encoder-decoder network was trained in an end-to-end fashion on both the indoor NYUDV2 dataset [225] and the outdoor KITTI dataset [86] and achieves superior and competitive performance in comparison with state-of-the-art [224].

As described above, one of the most efficient ways to deal with single image depth estimation being ill-posed is using Deep Neural Networks (DNNs) and pre-train it with labels. The trained network can be used to estimate the depth in new environment provided that the distribution of the test data is close to the train data. These category of methods are dense and fairly light considering the algebraic nature of these DNN universal functions.

1.4.2 Challenges and approaches: accuracy-generalization trade-off

Although deep learning based method are very successful in understanding the whole scene in a dense fashion and being fairly light considering the algebraic nature of this type of depth estimation, they suffer from 1) accuracy 2) poor generalization when the test and train distributions are not close. They are not able to accurately predict the scene with all of its details and when the test dataset distribution is far from the train dataset they might even fail. In addition, they rely on monocular cues which can be exploited for adversarial attack in security or safety systems [338].

On the other hand, traditional geometric methods, tend not to be sensitive to the above-mentioned issue since they essentially do not rely on any prior knowledge i.e. distribution

of any training data. Instead, they benefit from temporal information of sequences of images/videos or synchronous camera rigs to extract the depth in an unsupervised manner.

At the same time, explicitly modeling of a dynamic scenes as well as flexible objects in monocular depth estimation using traditional computer vision methods is a big challenge. The reason is lying on the inherent way of estimation: scene might changes in a flexible and dynamic way between two consecutive frames. It should be noted that deep learning based methods can handle them to some extend since they usually estimate the depth using single image, not completely though. The trade-off for single image depth estimation is loss of accuracy.

Considering weakness and strength of each of these two approaches, a hybrid methods which benefits from both good generalization of geometric methods by extending traditional geometric models ability to handle flexible and dynamic objects in the scene and interleaved it with deep learning networks to create a self-supervised training pipeline is the promising direction.

1.5 Measures to evaluate the performance of MDE

Assume D_i is the estimated depth map at the pixel i. Assume D_i^* is the ground-truth values of the depth map at the same spatial position. Assume that N indicates the total count of pixels which both the ground truth values for depth map and the estimated depth are valid. The evaluation metrics which are accepted and usually used by researchers in this field are

- **Absolute relative difference (Abs Rel):**

$$AbsRel := \frac{1}{N} \sum_{N} \frac{|D_i^* - D_i|}{D_i} \tag{1.3}$$

- **Squared relative difference (Sq Rel):**

$$SqRel := \frac{1}{N}\sum_{N}\frac{|D_i^* - D_i|^2}{D_i} \tag{1.4}$$

- **The linear root mean square error (RMS):**

$$RMS := \sqrt{\frac{1}{N}\sum_{N}|D_i^* - D_i|^2} \tag{1.5}$$

- **The logarithm root mean square error (RMS log):**

$$RMS\ log := \sqrt{\frac{1}{N}\sum_{N}|\log D_i^* - \log D_i|^2} \tag{1.6}$$

- **Accuracy based on a threshold:** is the percentage of the predicted pixels out of the total pixels which the relative error is less than a threshold.

$$\max\left(\frac{D_i}{D_i^*}, \frac{D_i^*}{D_i}\right) < \delta. \tag{1.7}$$

  The values of the threshold, δ, usually set to $1.25,\ 1.25^2,\ 1.25^3$.

In addition to the above-mentioned measures, Eigen et al. [67] introduced a scale-invariant error to express the relative error between points in the scene, independent from their absolute values. It is defined as

$$E(D, D^*) := \frac{1}{2N}\sum_{i=1}^{N}\left(\log D_i - \log D_i^* + \alpha(D, D^*)\right)^2 \tag{1.8}$$

where

$$\alpha(D, D^*) = \frac{1}{N}\sum_{i=1}^{N}\left(\log D_i^* - \log D_i\right) \tag{1.9}$$

is the α that minimizes E.

1.6 Datasets

Just like any other regression problem, datasets are critically important in developing and evaluating any depth estimation algorithm. There exist several well-known datasets. These datasets are summarized in Table 1.1.

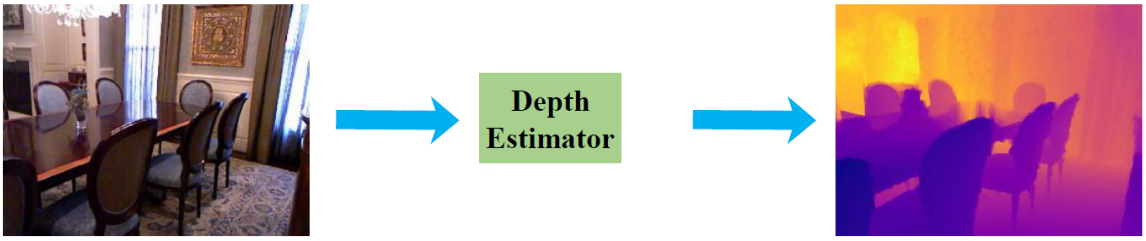


Figure 1.1: Casting single image monocular depth estimation as an estimation problem. A depth map is estimated using the corresponding RGB image.

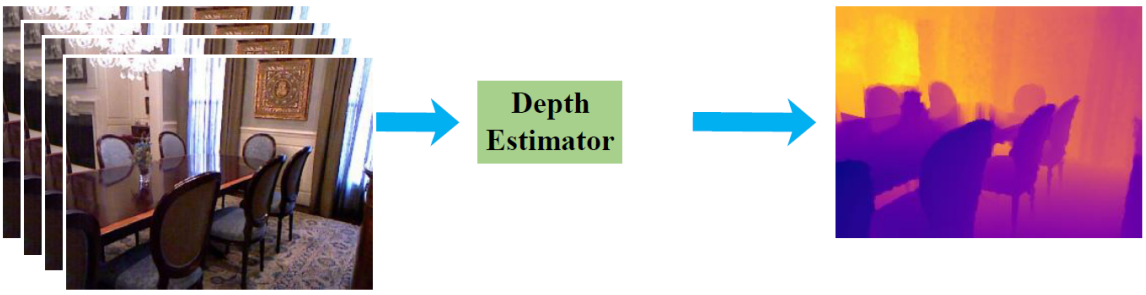


Figure 1.2: Casting monocular depth estimation from a sequnce of images as the estimation of depth map from the corresponding RGB image.

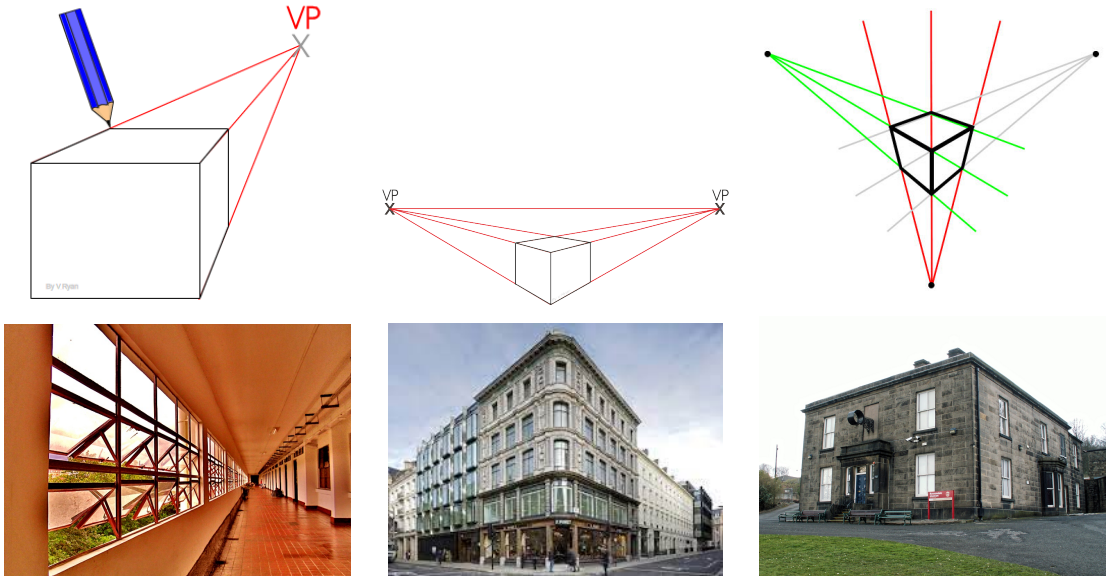


Figure 1.3: Comparison between 1-point, 2-points and 3 points linear perspective [328]. The vanishing point play an important role in MDE which will be discussed later.

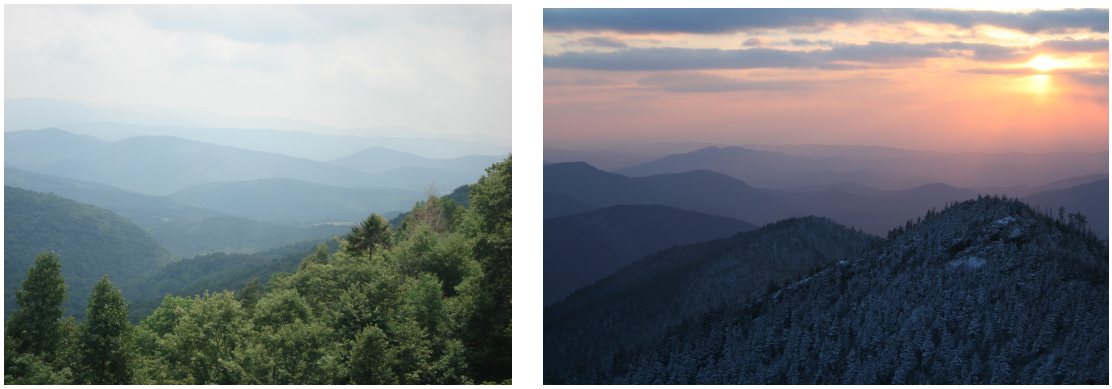


Figure 1.4: The aerial or atmospheric perspective; Left image: photographed in a nearly contre-jour condition (French for "against daylight"). Right image, the aerial perspective as a result of Rayleigh scattering. This is why when one look at mountains in Tennessee they see the far part of mountains in blue color [327].

Figure 1.5: Masolino da Panicale's St. Peter Healing a Cripple and the Raising of Tabitha (C.1423), this is considerede the first artwork known to utilize a consistent vanishing point [328].

Figure 1.6: Artists have been aware of atmospheric perspective. They tried to benefit from these monocular clue. In this picture, Dai Jin, ”Landscape in the Style of Yan Wengui”, Early Ming Dynasty (1368-1644); a Chinese landscape painting using ”atmospheric perspective” to show recession in space [327].

Figure 1.7: Comparison of edges and gradients in an RGB image and the corresponding depth map. Top-left: RGB image. Top-right: the corresponding depth map of the RGB image. Bottom-left: Laplacian of the RGB image. Bottom-right: Laplacian of the depth map. Figure from [224]. he cabinet on the left and the table surface are almost texture-less in the RGB image and need to have gradient in depth image while the wall with the brick texture mainly shows a gradient-less area in the depth map. Looking at the filtered RGB image and the filtered depth map suggests that most of the information needed to extract a depth map of a scene is near the geometric edges, i.e. edges which come from the 3D geometric structure of the scene. In addition to extracting geometric edges, one needs to get ride of the edges in the RGB image which mainly come from texture and colors and replace the texture-less area of the RGB image with deduced geometric structure in the depth map. This is a strong incentive to exploit the similarity of depth structure and the RGB image structure through some kind of attention mechanism to extract a better depth map.

Table 1.1: Summary of the available datasets which can be used in MDE. Table from [63].

Year	Dataset	Scenario	Sensors	Resolution	Type	Images	Annotation
2008	Make 3D [263]	Outdoor	Laser Scanner	2272×1704	Real	534	Dense
2012	NYUDV2 [275]	Indoor	Kinect v1	640×480	Real	1449	Dense
2012	RGBD SLAM [281]	Indoor	Kinect v1	640×480	Real	48K	Dense
2013	KITTI [85]	Driving	LiDAR	1238×374	Real	44K	Sparse
2015	SUN RGBD [277]	Indoor	-	-	Real	10335	Dense
2016	DIW [41]	Outdoor	-	-	Real	495K	Single Pair
2016	CityScapes [51]	Driving	Stereo Camera	2048×1024	*Real*	5000	Disparity
2016	CoRBS [325]	Indoor	Kinect v2	1920×1080 RGB, 512×424 Depth	Real	-	Dense
2016	Virtual KITTI [78]	Outdoor	-	1242×375	Synthetic	21260	Dense
2017	2D-3D-S [12]	Indoor	Matterport Camera	1080×1080	Real	71909	Dense
2017	ETH 3D [271]	In/Outdoor	Laser Scanner	940×490	Real	-	Dense
2017	Matterport 3D [37]	Indoor	Matterport Camera	1280×1024	Real	194400	Dense
2017	ScanNet [55]	Indoor	Structure Sensor	1296×968 RGB, 640×480 Depth	Real	2.5M	Dense
2017	SceneNet RGBD [211]	Indoor	-	320×240	Synthetic	5M	Dense
2017	SUNCG [278]	Indoor	-	640×480	Synthetic	45000	Dense
2018	Mega Depth [185]	In/Outdoor	-	-	Real	130K	Dense/Ordinal
2018	Unreal [207]	Outdoor	-	256×160	Synthetic	107K	Dense
2018	Safe UAV [208]	Outdoor	-	640×480	Synthetic	8137	Dense
2018	3D 60 [363]	Indoor	-	-	Synthetic	35995	Dense
2018	NUSTMS [332]	Outdoor	Radar	576×160 Infrared, 144×40 Depth	Real	3600	Dense
2019	DIML / CVL [47]	In/Outdoor	Kinect v2, Zed Stereo Camera	1920×1080 1280×720	Real	1M	Dense
2019	Driving Stereo [345]	Driving	LiDAR	1762×800	Real	182K	Sparse
2019	DIODE [301]	In/Outdoor	Laser Scanner	1024×768	Real	25458	Dense
2019	Mid Air [73]	Outdoor	-	1024×1024	Synthetic	119K	Dense
2020	Forest Environment [228]	Forest	Depth Camera	640×480	Real	134K	Dense
2020	Shanghaitech Kujiale [134]	Indoor	-	1024×512	Synthetic	3500	Dense
2020	Virtual KITTI 2 [30]	Outdoor	-	1242×375	Synthetic	21260	Dense

Chapter 2

Taxonomy of Algorithms in Depth Map Estimation

Depth map estimation can be bi-product of a bigger algorithm or the main goal of an algorithm or one of the main goals. In this chapter, different algorithms in which depth map estimation is involved are classified. While the algorithms can be classified based on several miscellaneous traits including, for example, the type of optimization terms or regression models vs. classification models or based on chronological order, one might decide to categorize the methodologies roughly into the following categories based on general mathematical/physical tools have been utilized for depth estimation. While doing so, it is also possible to describe the above-mentioned trait as well as single-view or multi-view MDE.

In addition, the methods will be described based on how they are trained whenever a learning based methods is under consideration. So these learning based algorithm classified into three categories, that is, Supervised, Semi-supervised, Self-supervised.

2.1 Monocular cues methods and optimization

In this paradigm, at least one of the goals is estimation of the depth map from a single 2D RGB image in inference mode (test or evaluation time) which is in fact the minimal sensory setup for the depth map estimation. During training, if there is any training, there might be more than one objectives optimized along with each other in one or more inter connected optimization loops. It is worth mentioning that the single image depth estimation usually called MDE as well which can be distinguished by context from monocular video depth estimation. In addition, monocular cues can help us to estimate better depth even if there is access to a video or sequence of images.

However, recovering the scene's depth from a single image is an ill-posed problem which means it is not mathematically possible to retrieve the depth map from a single RGB image uniquely. The reason is simple: going from 3D to 2D, one looses information which makes the reverse process impossible. So solving the problem requires additional priors, often referred to as monocular depth cues, like perspective, occlusion, object size, texture, etc., which briefly were talked about in the introduction. Although these cues might be exploited through methods based on learning with prior knowledge to disambiguate the solution, the first research in MDE was not a learning based method [118]. See Fig. 2.1.

It is known that the intrinsic images related to physical characteristics of a scene like depth, shadows, surface shape, provide critical information to depth estimation [18]. Exploiting the aforementioned fact, Kong and Black [152] cast the MDE problem as an intrinsic image estimation problem. They fuse the method in [144] with another procedure that to solve the MDE problem. See Fig. 2.2.

2.2 Learning-based depth estimation algorithms using engineered features

Torralba et. al. [293] designed the first learning-based algorithm. Their algorithm estimates absolute values of depth from monocular color images benefiting from the known size of objects in the scene and learning a structures' features from wavelet transform.

Jung et. al. [138] proposed a learning based MDE algorithm utilizing a Bayesian object classification. Object in the scene are classified as ground, cubic and plane and sky. Then the relative depth of each pixel is estimated using these four classes and models.

Saxena et al. [262] proposed a learning algorithm, which is fully supervised, to address the MDE. They segment RGB images into small patches. They utilize two different depth features, absolute features and relative features. Then they utilize a Markov Random Field (MRF) to model the depth of each patch in relation to its surrounding patches. Raza et al. [251] utilized a combination of engineered features and random forest regression and MRF to address MDE. Similarly, Liu et al. [189] designed an algorithm based on MRF that exploites semantic information to infere depth from a single image. The models are solved by utilization of the L-BFGS algorithm.

Ladicky et al. [161] show that simultaneous implementation of semantic labeling and depth estimation improve accuracy of both tasks. More precisely, they demonstrated that conditioning the first task on the depth values of the pixels help improving the performance of the classifier and vice versa.

Liu et al. [196] casted the MDE problem as a hybrid discrete-continuous CRF optimization that exploits the relation of super-pixels in different sections of a RGB image.

Karsch et al. [144] suggest utilization of a data-driven and non-parametric method to

adrress MDE. It is done based on the comparison of the image in hand with a dataset using GIST. Then, they utilize MRF to smooth the depths spatially.

Up to here the methods utilize engineered features as the main means to do MDE. However, deep learning has shown effectiveness of automatic dense feature extraction.

2.3 Utilization of deep-learning in MDE

Here works which utilize deep learning as the main mechanism to estimate depth in MDE problems are reviewed. The other methods discussed so far, like optimization or probabilistic and statistics and geometric computer vision methods, might be used to help improving the overall performance or even make MDE possible.

Application of deep learning in image classification was indeed a break-through in computer vision. As a result of such improvements, researchers in other areas of computer vision started to utilize deep learning networks in MDE problem as well. Here, the MDE problem based on deep learning algorithms are reviewed. As mentioned in the introduction of this chapter, the methods in this section are classified into three categories based on the way the training procedure depends on the depth's labels. See Fig. 2.3.

2.3.1 MDE, supervised learning

The general flow diagram of MDE problem using supervised learning is depicted in Fig. 2.4 part (a). In this setting, the CNN accepts one RGB image as input. During training phase, the corresponding ground truth depth map is compared with the output of the network. The error between the ground truth depth and estimated depth, i.e. the output of the network in the signal which is used by optimization algorithm to learn the parameters of the CNN network.

General supervised methods

The supervised deep learning based algorithm generally assume MDE problem is a regression. As far as we are aware, Eigen et al. [67] cast the MDE problem as a deep learning, supervised problem for the first time. In order to embed local and global scene information, they decided to utilize two CNNs. See Fig. 2.3 part (a). Moreover, they introduce a scale-invariant measure for the first time in addition to the available scale-reliant measures in evaluating the performance of their algorithm as well as in their optimization cost function. Their algorithm was a break through at the time which dramatically enhanced MDE reults on the KITTI [85] and the NYU [275] datasets.

It is possible to simultniously exploit continuous CRF and deep learning CNNs to performe a MDE. For example, Liu et al. [191] proposed a deep continuous conditional random fields to address the MDE.

The algorithms in [67], [191] rely on fully connected layers for depth prediction. Although, the layers have provide us with full receptive field, it has a too many parameters that results in a very slow inference procedure [67].

Laina et al. [162] proposed a deep learning residual network which is fully convolutional to address MDE problem. Their network comprises of two main module, one encoder and one decoder. See figure 2.3 part (b). The encoder is ResNet-50 [114] encoder without its final pooling and fully connected layers. In decoder, they design a novel up-sampling based on deep learning convolutional layers. In addition, they define the reverse of the Huber loss that perform well for depth map estimation based on the distribution of depth maps. They train their network in an end - to - end fashion. Their architecture does not depend on any post-processing algorithm like CRF and fully connected layers.

The proven effectiveness of CNNs resulted in utilization of CNN based pretrained encoders such as DenseNet-169 [124], ResNet-101/152 [114] or SENet-154 [122] to enhance MDE

performance. For example, Alhashim et. al. [9] proposed a densely connected encoder-decoder model based on DenseNet-169. Different from [162], they design a simple decoder structure which comprise of a bilinear up-sampling and convolution layers. With this deeper architecture, detailed augmentation and specific training strategies, the designed network creates more accurate estimations on the NYUDV2 [275] and KITTI [85] datasets. Mancini et al. [206] exploit optical flow to enhance depth estimation from single images.

To create a high quality depth map estimation based on automatic features from deep learning encoder, Lee et al. [168] suggest a layer which is called local planar guidance layer. They embed it in each decoding level. The output of the layers has equal size of the depth map. At the end, they are fused to build the depth map estimation.

Yin et al. [347] devise a constraint in the 3D space based on geometry to address the MDE problem. In fact, they suggest a geometric loss term. The designed loss function comprises of two terms. First a virtual normal term that compares the virtual normal vectors of the ground truth 3D reconstruction with the estimated 3D res-construction. Second they utilize weighted cross entropy as a mean of absolute depth supervisions. To sum up, the total cost function makes the MDE network estimates accurate and high quality 3D point cloud and depth map.

Hu et al. [123] introduce a module, an encoder-decoder architecture, benefits from a multi-scale feature fusion architecture and a refinement one. The multi-scale feature fusion architecture up-samples features maps from stages with different spatial sizes and then concatenate it channel by channel. Features from the multi-scale feature fusion architecture module and features from the decoder are combined with each other and then fused features are passed to the refinement module to estimate the final results. The main innovation of their work is a multi-term loss function, which incorporates depth errors, gradients errors and surface normal vectors errors. This work inspired Chen et al. [42] to introduce a Structure-Aware Residual Pyramid Network. The network utilizes scene structures in several scales to estimate depth map better. This module comprises three sub-modules, an

encoder module which outputs feature maps with several scales, an adaptive dense feature fusion module to fuse features maps and a residual pyramid decoder.

Fu et. al. [76] suggest instead of estimating continuous depth values, estimates the intervals. However pure classification does not consider ordered nature of depth values. So they suggested an ordinal regression instead of pure classification model to estimate the depth map. To this aim, they quantize the continuous depth values into a sequence of intervals. In fact, they transform the MDE problem into an ordinal regression problem. Since the estimated depth with larger values of ground-truth depth have larger uncertainty, the uniform quantization may lead into an overly tight loss for the larger depths. To address this issue, a space-increasing quantization method is introduced to quantize the depths. After discretizing depths, the ordinal regression loss is used to train the network.

However, considering constant intervals for all images and/or at all spatial points in the image is not the most efficient way of using computational resources in MDE. This is the incentive for Bhat et al. [21] to suggest to divide the range into adaptive bins, which the bins widths change in different images. This makes it possible that the network learns to adaptively focus on different depth values. Their contribution is Mini-ViT module comprising of four transformer layers [302]. It is utilizing a variant of Vision Transformer [64]. Different from Fu et al. [76] that uses the bin center which has the most probability as depth estimation, the depth estimation in [21] is the linear combination of bin centers which is weighted by the probabilities. Hence, this method results in a smoother estimation. See Fig. 2.5 and 2.6.

Recently, omni-directional cameras are becoming increasingly attractive to researchers. The creation of depth map using a single 360° image [363], [313] has been explored by researchers. Different from normal cameras, the omni-directional cameras have got a dramatically larger field of view which makes it possible to capture the 360° surrounding scene. The first attempt to extract the depths using an omni-directional RGB image is OmniDepth introduced by [363]. They contribute to the field by creating a dataset comprise

of 360° RGB and their counterparts depth map. Since capturing such ground truth depth map is difficult, they created the dataset out of available 3D datasets. They innovate a two-input model which combines cubemap projection and equirectangular to infer the 360° depth map from monocular 360° RGBs.

Other than CNNs, Recurrent Neural Networks are also utilized to address the MDE problem. RNNs are able to capture the temporal behavior in sequential data in time since they model a difference equation, i.e. they are a dynamic model. See Fig. 2.7. As an extension of the regular RNNs, the long short term memory is able to learn long term reliance in time between different sections of the input data. See Fig. 2.8.

Kumar et al. [53] proposed an encoder decoder architecture based on a convolutional LSTM (ConvLSTM) module. They try to extract the depth map in a sequnce of video frames exploiting the spatio-temporal information. The encoder comprises of a several ConvLSTM stages. The decoder consists of interleaved deconvolutional and convolutional layers. ConvLSTM layers, each have internal states which are related to the length of video.

Zhang et al. [354] try to benefit from both temporal and spatial information to address MDE problem. They utilize both GAN and ConvLSTM. Moreover, a temporal consistency loss is created to keep the consistency between different frames in the sequnce of frames. Both the spatial and temporal loss are used to train the model in an end-to-end fashion.

Similar utilization of RNN, more exactly ConvLSTM, in video MDE problem is introduced by [320].

Monocular depth estimation through classification

The depth values in different pixels in one RGB image, might have statistical distribution which are different. As a result, the problem of depth estimation can be cast as a classification task by segmenting the scene [33], [175], [181], [44], [365].

Li et al. [175] suggest a three stage depth estimation methodology. First, they define the MDE as a dense labeling task which makes the work different from the traditional regression MDE. Second, they combine different side outputs from dilated convolutional neural network to benefit from the depth features in a multi scale fashion, which gives their algorithm scale perception. Third, they use a weighted sum inference which transforms the quantized depth probabilities to continuous depth values. Consequently, they enhance the robustness of the overall algorithm and decrease the quantization error.

Following the mentioned works of [33], [175], [181], Zou et al. [365] address the MDE problem as a classification, too. However, they consider probability distribution in training phase. The innovation of their work is a mean-variance loss function which comprises of a variance part and a mean part. The mean loss penalizes the error of the means of the ground-truth depth map and the estimated depth map distribution. The variance loss is creates a sharper estimated depth map distribution. The mean-variance loss as well as the softmax loss provide the supervision in training.

Moreover, not only MDE can be fromulated as a solely regression or solely classification problem, but also it can be formulated as combination of both paradigms together[186], [279].

Multi-task learning methods

The MDE problem and the other problems in computer vision like surface normal estimation, semantic segmentation are able to improve performance of each other if they are done simultaneously [66], [319]–[1]. For example are similarities between the depth maps and the semantic maps [259]. Even designing a network suitable for more than one task can be beneficiary to each tasks, although they are not done in a simultaneous fashion. One of the first examples is Eigen and Fergus [66] who proposed a network architecture to address surface normal estimation, MDE, and semantic segmentation problem using one architecture. The network can learn to do each of the three tasks if one changes the loss function and the output layer.

Wang et al. [319] and Jafari et al. [130] proposed an architechture to address simultanious estimation of the semantic labels as well as the depth maps.

The afore mentioned algorithms [66], [319], [130] need dense ground truth for semantic segmentation labels and depth maps values. This makes them difficult since gathering those information is not an easy task to do. So Gurram et al. [103] address this issue by exploiting information from two heterogeneous datasets in the task of training a CNN which estimates the depth map. The overall learning comprises of two stages. First, a training procedure is used for pixel level semantic and depth classification. In the second step, the regression layers refines the results.

Qi et al. [245] proposed a Geometric Neural Network, GeoNet, simultaneously learns surface normal vectors and depth maps in a monocular fashion. GeoNet comprises of a network to estimates normal vectors from depth maps and another network to do reverse. Iterative usage of these two networks converges to both depth map and the surface normal vectors.

Hesieh et al. [121] suggest an architecture based on YOLOv3 [252] which does object detection as well as MDE.

Abdulwahab et al. [1] proposed a paradigm for both 3D pose estimation and depth map estimation utilizing a CNN as a regressor and a GAN block. The GAN module does feature matching which makes it possible to construct the depth map. The regression module uses the depth map to estimates the poses in 3D.

Real-time supervised monocular depth estimation

The above-mentioned methods are computationally expensive. However, there are several real-time applications that researcher do a trade-off between the quallity of the depth map and the speeed of inference [280], [329]. Depth Net Nano [318] is another example of these category of networks.

Learning algorithms which are supervised need ground truth depth maps in training phase. Although they are high quality methods, gathering the ground truth is difficult due to cost of depth sensors like RGBD cameras and LIDARs as well as calibration and synchoronization difficulties. As a result, learning algorithms which are self-supervised are becoming a trend in research [83]-[361].

2.3.2 MDE, self-supervised learning

Self-supervised learning algorithms use two or more sequences of RGB images as input and cast the problem as an image reconstruction problem. Under this situation, the depth maps are biproduct of the reconstruction process. The flow diagram of unsupervised learning algorithms is depicted in Fig. 2.4 part (b). These methods do not need ground-truth. However, the accuracy rate is lower than their supervised cousins methods.

General unsupervised algorithms

Garg et al. [83] proposed the first self-supervised learning algorithm in MDE. First, a pair of close RGB images which have known ego-motion The image reconstruction loss is the supervision signal to train the weights of depth network, which estimates the depth out of the corresponding RGB image. Moreover, the point registration between the two RGB images is another supervision signal in this algorithm. Similarly, Godard et al. [91] design a self-supervised algorithm that uses a left-right consistency loss that exploit aliened stereo RGB image pairs and epipolar geometry constraints.

Both [83] and [91] need calibrated stereo RGB images to train their networks. So Mahjourian et al. [205] relax this limitation enforcing the consistency between camera motion and the depth map of one of the images. Similar works are done to design self-supervised MDE in [294], [203], [356], [71].

Guizilini et al. [98] exploit semantic information to learn depth estimation. The method is based on an self-supervised paradigm [96]. It comprises of two models, one which

estimates the depth, another does semantic segmentation. At training time, the depth estimation model learns its parameters while the other model is frozen.

Also, Johnston et. al. [136] proposed a discrete disparity volume and similar self-supervision methos to estimate more clear depth map along with pixel wise depth uncertainties.

Multi-task learning based methods

Zhou et al. [361] address the problem of learning the camera motion and the depth maps from monocular videos. The method uses a depth network [210] and a pose network. They use reconstruction error as the supervision source. Scale ambiguity is a problem here.

Inspired by [361], Prasad et al. [243] exploit epipolar geometry as the constrains to learn both depth and ego-motion. Different from [210] they used epipolar geometry to weight the pixels and guide the training.

Klodt et al. [150] change [361] in three ways. First, a structural similar loss helps to robustify the brightness constancy loss. Second, an explicit model of confidence makes sure correct prediction of each pixel brightness distribution. Third, a SfM algorithm [222] is used to supervise the depth network training.

Vijayanarasimhan et al. [306] proposed "SfM-Net," a geometry-aware algorithm to jointly estimate depth, camera motion and dynamic object segmentation. The network uses photometric error as source of supervision.

Dai et al. [56] design the same algorithm as in [306] for monocular video. The motion model is 6 degrees of freedom in this work instead of 2D/3D optical/scene flow.

Learning based SLAM algorithms usually assume that the scale of CNN-based MDE and relative pose can be consistently learned between all input frames. This assumption adversely affects the performance in situations where the change of size of translation

of camera is large. In order to address this issue, Bian et al. [22] suggest a geometry consistency loss.

Also, Zhao et al. [358] separately estimates a scale for the learning of the pose and the depth.

Zou et al. [366] design a self-supervised algorithm to both learn optical flow and depth from monocular video. They design a cross-task consistency loss in addition to the photometric and spatial smoothness loss as the sourse of supervision.

Yin and Shi [348] design an algorithm that learns the depth map, and the camera pose as well as the optical flow at the same time. First, they utilize statc scene methods. Then they suggest a non-rigid motion refinement module to deal with the dynamic objects. Also they proposed an adaptive geometric consistency loss which addresses the texture-less areas as well as the occlusions.

Ranjan et al. [248] suggest an algorithm that estimates the camera motion and the depth map as well as the optical flow and the motion segmentation. They utilize a Competitive Collaboration (CC) learning method.

Learning methods using adversarial paradigm

Not only view-synthesis or photometric reconstruction error, but also Generative Adversarial Networks (GANs) paradim can help to do unsupervised MDE [8], [212], [239]–[10]. GANs have two part a generator and a discriminator (Fig. 2.3 (d)). The two parts can be used to create unsupervised learning algorithm. The discriminator distinguishes between the real images and the synthesized ones, though.

Aleotti et al. [8] utilizes GAN to address unsupervised MDE for the first time. The generator is designed to estimate a depth map from the RGB image and generate a warped synthesized image. The discriminator then discriminates between the input real image and warped image and. The generator is obligated to estimates depth maps because of wrapping

process. Similarly, Mehta et al. [212] proposed a MDE using GAN paradigm using stereo synthesis. Wang et al. [307] combines GAN with direct visual odometry. In this way, he was able to design an unsupervised dense MDE algorithm. Almalioglu et al. [10] suggest an unsupervised MDE algorithm based on GAN and recurrent paradigm.

Real-time unsupervised monocular depth estimation

Although all the state-of-the-art algorithms produce promising results in unsupervised paradigm, they are very heavy and complex which makes it impossible to use them in many robotics applications where lower power and speed are important. To address these issues Poggi et al. [240] utilizes a small encoder and multiple small decoders in a pyramidal. He is able to achieve almost real-time performance on a i7-6700K CPU. Liu et al. [195] design MiniNet based on DepthNet. It is a small network trained in an unsupervised manner on a monocular video. The algorithm reaches real-time speed on a Nvidia 1080Ti GPU.

2.3.3 MDE, learning in semi-supervised paradigm

Self-supervised algorithms do not rely on the ground-truth. However, the performance is bottle-necked by SfM reconstruction performance. It motivates semi-supervised methods [47], [160], [353], [11], [99], [287], [132]. These methods exploit limited number of ground-truth and the rest of training data without ground-truth. In this way, they are able to enhance the performance of the MDE. See Fig. 2.4 (c). At the beginning, the model is supervised with the available limited number of ground-truth. Then the trained network is exploited to infer the depth maps of the rest of the training data-set. At the end, the inferred depth maps as sudo labels as well as the limited number of ground truth are used to train the model just like the first step.

General semi-supervised methods

Kuznietsov et al. [160] proposed semi-supervised MDE for the first time. He essentially combines self-supervised and supervised cost functions. The supervision comes from the reconstruction of stereo images as well as sparse depth pairs. Amiri et al. [11] develop a semi-supervised MDE algorithm based on [91]. They include sparse ground-truth, LiDAR, depth as additional signal while unsupervised signal is stereo frames. Guizilini et al. [99] is another semi-supervised algorithm in MDE. Differently, Ji et al. [132] suggest a GAN based semi-supervised algorithm.

Joint Semi-supervised tasks

Ramirez et al. [353] suggest a semi-supervised algorithm which jointly estimates depth and semantic segmentation. Their model consist of a semantic decoder head, a depth decoder head, and an encoder which is shared. The training of semantic segmentation head is supervised. However the MDE sub-model trained in an unsupervised manner via re-projection cost function. Similarly Yue et al. [352] suggest a semi-supervised algorithm to estimate depth map benefiting from semantic segmentation. Again the depth training algorithm is unsupervised part.

Tian et al. [287] suggest a semi-supervised MDE based on a depth model and a confidence model. The confidence model uses the output of the depth model and feed it along with the corresponding RGB image into the confidence sub model. The confidence sub-model estimates a confidence map that can be used to train the depth sub-model.

Student-teacher learning paradigm is what Cho et al. [357] exploited in their semi-supervised algorithm. They utilize a deep stereo matching model [233]. The model is trained in a supervised manner. and used to train a small student model. They assert that the small model performs better in this way in comparison with the trained teacher model.

Semi-supervised algorithms can be utilized not only when a small set of the ground truth depth is available but also when a semantic maps or sparse depth maps are available. It usually delivers better results than the supervised algorithms.

2.3.4 MDE, domain adaptation

With the advent of advance computer graphics, synthetic data-sets became available. The readily available synthetic datasets can be used to train depth models in a supervised manner. However, the distribution of data in real scene and the synthetic datasets are different. The models which are trained on these synthetic scene do not generalize well to the real world scenes. This is called domain gap. To reduce the adverse effect of domain gap in MDE, researcher suggests domain adaptation algorithms. See Fig. 2.4 (d).

Domain adaptation via fine-tuning

The first idea came to researchers mind was training on synthetic datasets and then fine-tune on a small set in target domain with ground truth lasbels. DispNet [210] is the first one that utilizes this approach. However, it is possible to fine-tune the model which is trained on the synthetic dataset on another supervisory signal like stereo depth estimation. Guo et al. [101] is first reaseracher who does that. Their accuracy is better than [67], [91], [361], and [160]. This approach is effective provided that there are enough data to fine-tune the model [289].

Domain adaptation via data transformation

Although the domain gap can be solved using fine-tuning technique, it is not the only way to make the domain gap smaller. Another way is using some transformation to make the data in the two domian become more similar to each other. Atapour-Abarghouei et al. [13] design a MDE paradigm based on GAN to make the distribution of the synthetic and the real dataset become similar. Similarly Zheng et al. [360] design an algorithm based on

GAN to translate the synthetic dataset into real world images. Their model is trainable in an end to end manner.

However, [13], [360] do not take into account the geometry of the scene. So Zhao et al. [357] incorporate the epipolar geometry to develop a geometry aware domain adaptation based on style transform. These algorithms based on the transformation of the data are robust to differences between different domains. However just like most of robust approaches it comes with a side effect which is lower accuracy [289]. Also, variations in illumination or saturation of the images might degrade the performance of the transformed images. Thus the overall MDE accuracy [135].

2.4 Geometric computer vision methods

Geometric computer vision methods defines the relation between observations in the images and the 3D scene and utilizing camera models and machine learning approaches to estimate the 3D model for the scene under observation. Finding the 3D model is called mapping and finding the the orientation and position of the camera in relation to the scene is called localization or ego-motion estimation. The methods are divided into two set of algorithms.

- Structure from Motion, SfM
- Simultaneous Localization and Mapping, SLAM

They both estimate the mapping or structure of the scene as well as the position of the camera with respect to it. The difference is that the SLAM algorithms are designed in an online fashion. These algorithms are fed by the input images as a sequence of the images and most of the times they need to be real-time as well. However the SfM algorithms do not need to be online or real-time necessarily [297]. In addition, a complete Visual SLAM usually relies on different type of sensors' data like camera and inertial measurement units (IMU) and miscellaneous sub modules like visual odometry or visual inertial odometry,

bundle adjustment optimization, loop closure and re-localization and dense mapping. So a light SfM might be one internal part of a SLAM algorithm.

2.4.1 SfM

Assume that there is a set of images of a scene from different point of view. First, a set of usually engineered features like SURF, SIFT, Harris, AKAZE, are produced from each gray-scale image. Next, the features which are representing the same 3D points are registered to each other. At this step, often a robust method to outliers, like RANSAC, is applied. In this way, the points are tracked and using computer vision geometry the 3D point cloud is produced. At the end this point cloud is converted into a depth map. Prakash et al. [242] is an example of the utilization of SfM for sparse MDE.

Ha et al. [104] design a SfM using small motion, i.e. SfSM, that is based on plan sweep paradigm for MDE. Feature extractor here is the Harris corner detector and the Kanade-Lukas-Tomashi algorithm to match them. At the end, the plane sweeping method is used to produce the structure of the scene. This method is very slow. So Javidnia et al. [131] suggest to use the ORB features instead. This change make the algorithm faster. However, the ORB features are sensitive to the amount of texture in different part of the scene. In low texture area it creates low accuracy results.

2.4.2 Visual SLAM

SLAM consists of two interleaved problem, one is finding the structure of the scene, called mapping, and the other is finding the position of the camera with respect to the scene, called localization. The sensor that is used the most in the SLAM algorithms are cameras. These SLAMs algorithms are Visual SLAM. Visual SLAM categorized into stereo camera, monocular camera, RGB-D camera and event camera, etc. based on the camera setting.

Monocular camera

Monocular Visual SLAM has scale ambiguity [107] and it needs to be initialized. The algorithm is afflicted by drift issue as well.

Stereo camera

Stereo camera setting which means using two or more camera at the same time which their physical positions are fixed with respect to each other. This method solves the scale ambiguity at the cost of difficult calibration and high calculation costs.

RGB-D camera

Depth cameras are called RGB-D. They are able to estimate the depth for each pixel using active methods. More precisely, they emit structured-light pattern and then build the map using IR stereo cameras. At the end they synchronize the depth map and the color image. All of these are done internally on the device hardware. Some have IMU inside as the extra sensor. The others are time of flight cameras (TOF). TOF cameras calculate the time that the emitted laser beam needs to travel the distance of the real world point to the camera.

Event camera

There are a category of cameras, called event cameras, which record variations in each pixel brightness asynchronously instead of recording frames at fixed frame rates. Event cameras are different in dynamic range (60 dB to 140 dB), different resolution, and they consume low power. Also they do not get motion blur. These traits makes them a good candidate in fast moving scenes.

Classification based on features

The SLAMs algorithms can be categorized into direct methods and feature based method. Direct methods use the intensity in the images directly. CNNs are direct method in this sense. Direct methods are able to estimate a semi dense or dense struture for the scene. On

the other hand, one might extract sparse engineered features, like SURF, SIFT, AKAZE, etc. and matches them in different images. Then use the registered points along with the computer vision to estimate the depth of the registered points in 3D. This is called indirect method. Indirect methods usually estimate sparse 3D clouds.

Visual SLAM which are sparse

- MonoSLAM(monocular): Extended Kalman Filter (EKF) is an online estimator. This makes them a good choice for SLAM algorithms. mono-SLAM [57] is the first SLAM algorithm which is based on monocular camera setting and works in real-time. The algorithm uses EKF.

- The first SLAM which separate the mapping and the tracking loop is Parallel tracking and mapping (PTAM). The paradigm is a monocular camera setting that does Bundle Adjustment for better accuracy and consistency. It also exploit the key frames concept for robustness. Later on they added relocalization to the algorithm.

- ORB-SLAM is a monocular camera setting which implemented using three threats. 1-Tracking, 2-Local Mapping 3-Loop Closure. They extend it to ORBSLAMv2 for RGBD and stereo rigs. CubemapSLAM the monocular fish-eye cameras setting of ORB-SLAM. It also uses IMU for scale estimation.

- ENFT-SfM is another monocular camera setting SLAM. Its distinctive trait is its ability to track points between one or more videos. It was extended to ENFT-SLAM to be able to handle large scale data.

- OpenVSLAM has the ability to accept mono, stereo, RGBD camera settings. It is an indirect method. The distictive feature of OpenVSLAM is its ability to use arbitrary camera models one may use.

- TagSLAM is a SLAM algorithm that is implemented using AprilTag fiducial markers. Te algorithm is the sub-module for the GTSAM factor graph optimizer.

Semidense visual SLAM

- LSD-SLAM is a monocular camera setting based on a novel direct tracking paradigm. It utilizes Lie Algebra and it is a direct method. Later on the algorithm was developed to omni-directional and stereo camera settings.

- SVO is a monocular camera setting SLAM algorithm which is a Semi-direct Visual Odoemtry. The algorithm utilizes sparse model-based image alignment to achieve speed. Later on, they imrove it so that it can use several cameras or catadioptric or fisheye cameras. CNN-SVO the SVO algorithm which is equiped with a CNN for single image MDE.

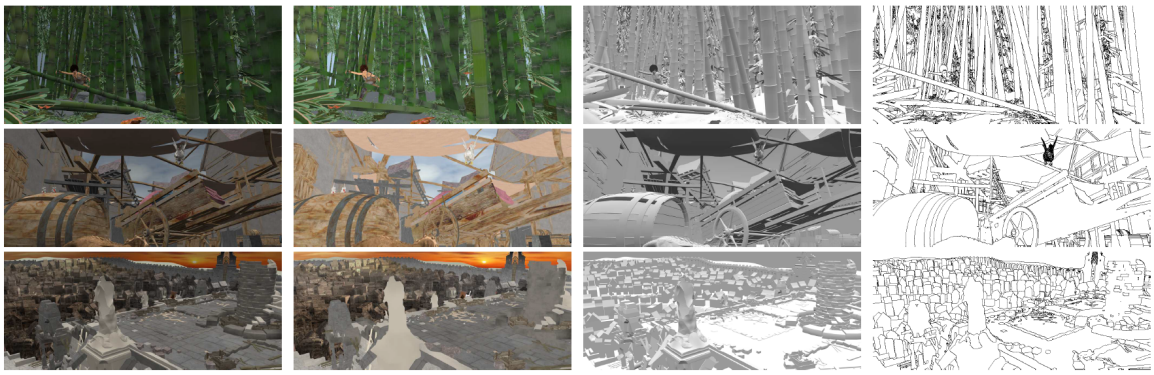
- Direct sparse odometry (DSO) is another monocular camera setting. The algorithm is a sparse direct visual odoemtry without detection and description of feature point.

Dense visual SLAM

- Direct tracking and mapping (DTAM) is a direct method in monocular SLAM category. It reconstructs impressive dense 3D model in real-time. The algorithm minimizes a spatially regularized energy function which is global. The optimization is non-convex and works directly with intensity of images for the first time. As a result it is called direct method.

- MLM SLAM is a monocular camera setting SLAM that estimates dense 3D model in an online fashion. The algorithm does not need any GPU. The novelty of this algorithm is its multi-resolution MDE paradigm as well as spatial smoothing procedure.

Figure 2.1: first figure from right: definition of the incident (i), emittance (e) and phase angle (g). Second and third figures:pictures of a nose with superimposed characteristic solutions and contours. Shape determined from the shading (not-intensity contours). Figure from [118].



(a) From left to right: RGB, abledo, shading, boundaries

(b) Example contour detection

Figure 2.2: Contours of surfaces estimated using shading and albedo. Figure from [152]

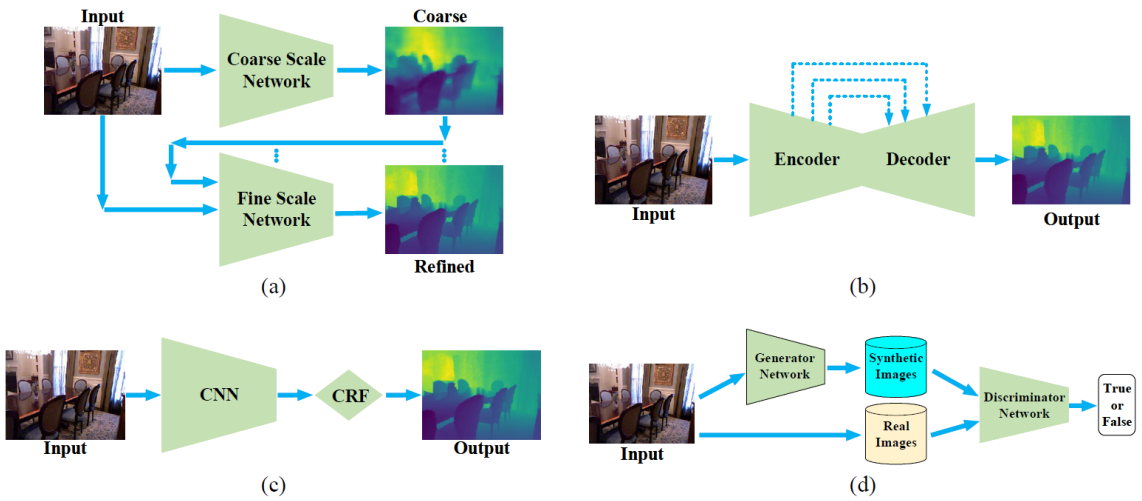


Figure 2.3: Classification of different architecture in MDE. (a) Multi-scale MDE methods [67], [66], (b) Encoder-Decoder architecture [162], [123], [204], [45], [42] (c) Combination of both CNN and CRF [33], [176], [116], (d) GANs [8], [212]. Figure from [63].

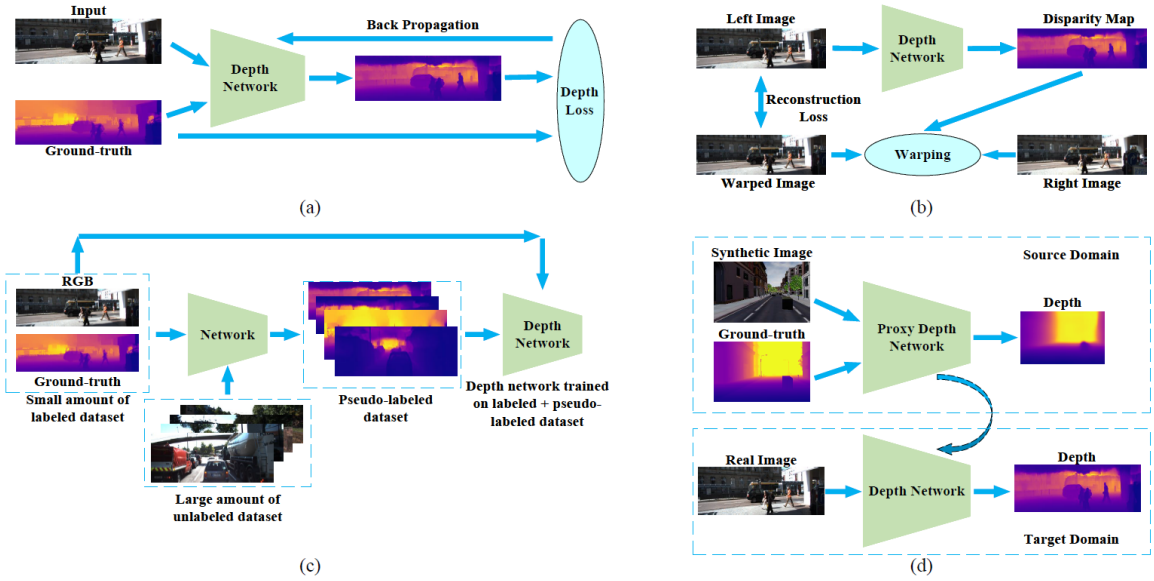


Figure 2.4: The general structure of deep learning models in MDE. In (a) general supervised learning algorithm is depicted. In (b) general unsupervised learning algorithm is depcted. In (c) general semi-supervised learning algorithm is depcted. In (d) Domain adaptation methods are depicted. Figure from [63].

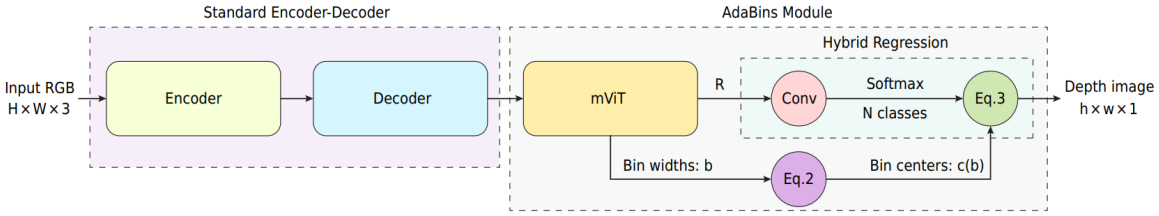


Figure 2.5: Adabins architecture [21]. Figure from [21].

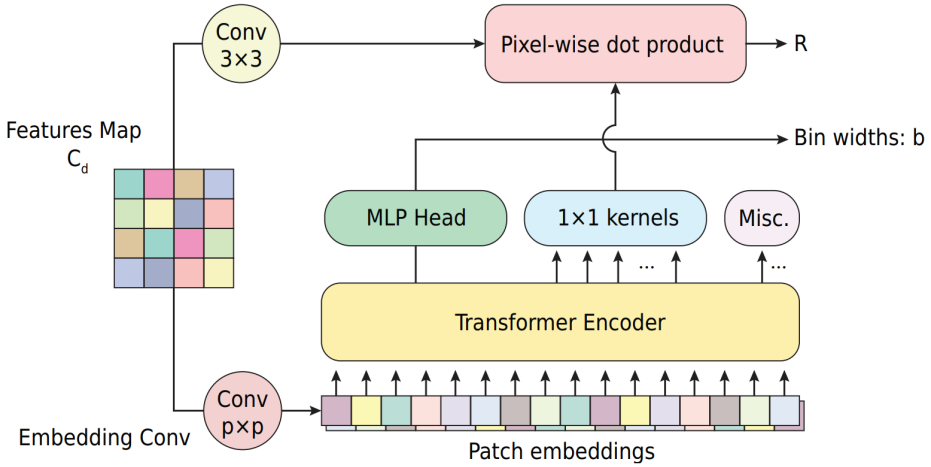


Figure 2.6: The mini-ViT block [21]. Figure from [21].

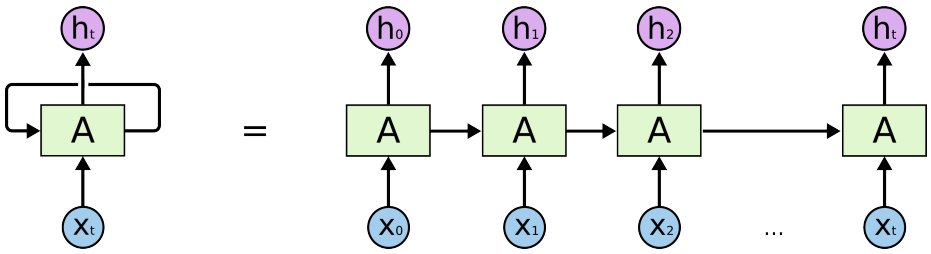


Figure 2.7: An unrolled recurrent neural network. Figure from [50].

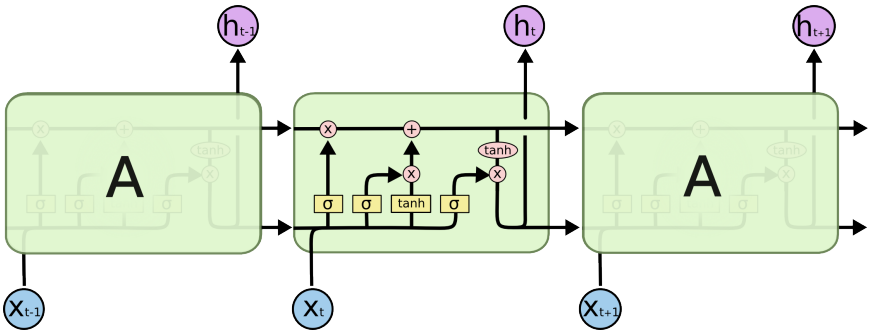


Figure 2.8: The repeating module in an LSTM contains four interacting layers. Figure from [50].

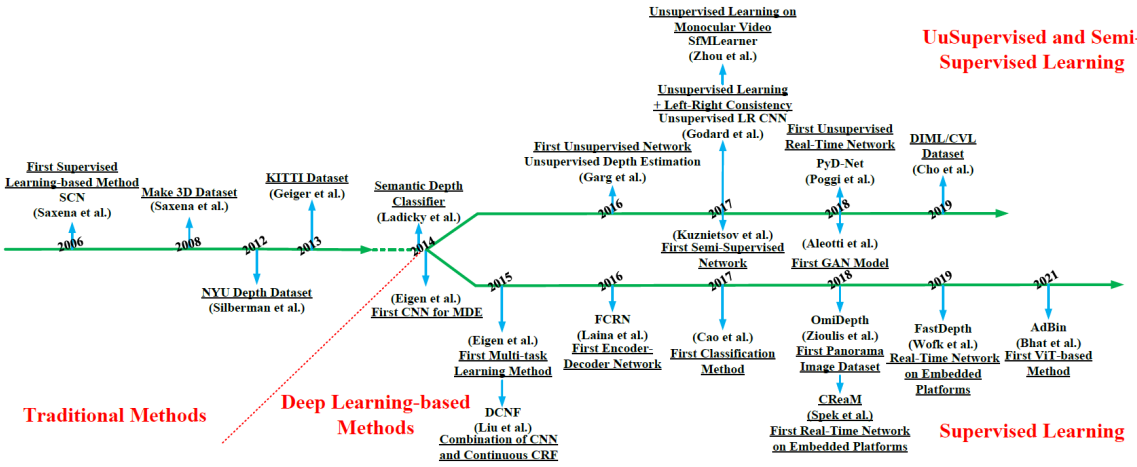


Figure 2.9: The Milestones of Monocular Depth Estimation. Figure from [63].

Chapter 3

Single Image Monocular Depth Estimation Using Adaptive Geometric Attention

3.1 Motivations for adaptive geometric attention in single image MDE

The minimal sensory setup for depth estimation is to use a single monocular image. However it is an ill-posed problem. That is, it is not mathematically possible to uniquely estimate the 3rd dimension (or depth) from a single 2D image. Hence, additional constraints need to be incorporated in order to regulate the solution space. Here the idea of constraining the model by taking advantage of the similarity between the RGB image and the corresponding depth map at the geometric edges of the 3D scene for more accurate depth estimation is explored.

Human is able to utilize attention to understand the local similarity between an RGB image and its corresponding depth map easily or even one can be deduced from the other. Human is able to do that since most of the information needed to extract the depth map of an

RGB image is near the geometric edges, i.e. edges which comes from the 3D structure of the scene. Here it has been proposed to give a convolutional neural network the ability to deduce the local geometric structure of the 3D scene in an RGB image using guidance from the corresponding depth map. However, the depth map is not available at the evaluation time. Instead, a general light-weight adaptive geometric attention module that uses the cross-correlation between the encoder and the decoder as a measure of this similarity has been proposed [224]. More precisely, the cosine similarity between the local embedded features in the encoder and the decoder at each spatial point is exploited.

The proposed module along with the encoder-decoder network is trained in an end-to-end fashion and achieves superior and competitive performance in comparison with other state-of-the-art methods [224]. In addition, adding the module to the base encoder-decoder model adds only an additional 0.03% (or 0.0003) of the total parameters of the network. Therefore, this module can be added to any base encoder-decoder network without changing its structure to address any task at hand.

The idea has been explained using the example shown in Fig. 3.2 that compares an RGB image and the corresponding depth map. The cabinet on the left and the table surface are almost texture-less in the RGB image and have gradient only at geometric edge locations in the depth map. On the other hand, the wall with the brick texture mainly shows a gradient-less area in the depth map but a lot of gradient in the RGB image. Looking at the high-pass filtered RGB image and the depth map suggests that most of the information needed to extract a depth map from a scene is near the geometric edges, i.e., edges in the RGB image which come from the geometric structure of the 3D scene. However, to extract the geometric edges, one needs to first remove the edges in the RGB image which mainly come from texture and color changes and replace the texture-less area of the RGB image with the deduced geometric structure in the depth map.

3.2 Related works

The ability of CNNs to work as a regressor has made them a good candidate for depth estimation. However, compared to estimation of the exact depth of a single point, it is easier to estimate its depth range [33, 76] and formulate the depth estimation as a pixel-wise classification task instead.thework benefits from both methods.

3.2.1 Depth estimation with (geometric) constraints

Deep learning methods have been proven to be effective in depth map estimation. However, they lack local details in 2D and they are often highly distorted when the maps are projected into 3D. In this case, One can also improve depth estimation using some kind of (geometric) constraint. While [123] tried to solve these issues by fusing multi-scale features, [347] exploited the virtual normals of virtual surfaces to estimate the depth map in 3D scene robustly. By the same token, [179] proposed a two-streamed CNN that predicts both depth and depth gradients and then fusing the outputs together into a detailed depth map. Another example of two-streamed CNN is GeoNet [245], which jointly predicts depth and surface normal maps from a single image. Similar to [66, 347, 245] which exploit geometric constraints, [168] assumed local planar for every local patch to guide depth prediction more effectively.

Intuitively, neighboring pixels with similar appearances should have similar depth estimation and major depth changes usually lie in the vertical direction in outdoor scenes. This constraint was utilized in [80] for single image depth estimation.

3.2.2 Super-resolution depth map estimation

Another category of works which are closely related to this work are guided depth super-resolution or GDSR. These category of algorithms reconstruct a high-resolution depth map out of a low-resolution depth map using the corresponding high-resolution RGB image. An example of this category of algorithms is [359]. The author suggests a discrete cosine

transform network which does three tasks. First of all, the network rebuilds the multi-channel high-resolution depth features to be used in solving the channel-wise optimization from image domain. Second, this feature extraction is done using a semi-coupled feature extraction module. Last, they creates an edge attention mechanism to emphasis the the contours for up sampling in a guided fashion.

3.2.3 Depth estimation in relation to segmentation

Depth estimation and semantic segmentation symbiosis represents one of the closest relationship in deep learning tasks. Some works have tried to exploit one to help improve the performance of the other or both at the same time [180, 316, 98, 66, 133, 115, 156, 316]. However, the performance is not the only incentive for this symbiosis. For example, [149] exploits semantic guidance to solve the dynamic object problem in monocular depth estimation.

Improving depth estimation using semantic segmentation can be interpreted as attending to the objects and their borders instead of all pixels just like in [312, 133]. While pixel-wise visual attention maps have shown their effectiveness [145, 312] suggested an object-level attention model for autonomous driving.

3.2.4 Depth estimation based on attention and transformers

Attention mechanisms have been used in depth estimation works previously. Most of the works are based on [322, 136] which in turn borrowed the idea from natural language processing (NLP) [302]. The suggested dot products and matrix multiplications usually try to find correlation between different spatial parts of tensor features [283, 136, 302, 346, 44]. The problem with these operations is they are computationally expensive where optimization is made more difficult due to lots of multiplication operations involved. Similar to [190, 335, 33], the authors in [336] employed a continuous CRF to fuse multi-scale information derived from a CNN. Different from the past works, they imposed structural constraints on an estimated attention map to estimate depth. Attention fusion was

also used in [106]. In [2] the authors, inspired by neural machine translation, introduced a CNN scheme which exploits forward and backward attention mechanisms. [136] used a self-attention context module to explore the inference of similar disparity values at non-contiguous regions of the image. Exactly the same mechanism was also adopted in [209]. Very similar to the above-mensioned works is [351]. They benefits from vision transformer in the encoder and fully connected CRFs as decoder. The fully connected CRF is essentially a graph model which is possible since they divide the whole spatial size into different Windows to reduce the computational complexity.

While attention and geometric constraint are beneficial for depth estimation, combination of both can be exploited to improve depth estimation [127, 314]. [100] tried to use attention mechanism to improve monocular depth estimation as well. Different fromthework, their spatial attention mechanism is separated from their global context module while ours combines the two stages in one light-weight and local module with different operations, i.e., sensitivity-enhanced geometric similarity in embedded Euclidean space.

Attention can be easily exploited in loss function since the ground truth depth is available when training the network. Having this in mind, [133] has tried to benefit from an attention-driven Loss that adjusts the backpropagation flow accordingly.

3.3 Proposed method

The structure of my model and the optimization as well as an in-depth discussion about the proposed module are discussed in this section. As discussed in Sec. 3.2, depth estimation can be defined as a regression problem or a classification problem. The model along with its cost functions are chosen from [347] as the base model which uses both classification and regression at the same time. Then the proposed attention module is integrated into the base model for performance improvement. The addition of the module imposes a minimal change to the base model in the sense of computational cost and only adding few additional parameters to the network model.

Like any other regression and/or classification problems, there are two aspects of the method which contribute to the quality of the estimation, namely, the model and its structure, and the cost function and the optimization method. In the following, the both aspects are elaborated.

3.3.1 Model

It is desired to guide the encoder to shape the RGB features using the depth map for better depth estimation at each spatial point. However, at prediction time the depth map is not available. Instead, the local cross-correlation of the embedded encoder and decoder features are used as the local similarity measure at each spatial point. The eventual criteria for this guidance is the sensitivity enhanced absolute value of the cosine similarity between the local embedded features at every spatial point of the encoder and the decoder. By enhancing the sensitivity, it has been tried to make any non-zero correlation between the encoder and the decoder features at each spatial point more effective. The similarity measure is absolute and normalized version of dot product (dot product is the conventional attention technique) which means more constraints are imposed on the network to regulate the solution space better.

The model and cost functions are adopted from [347] asthebase model. Then the proposed adaptive geometric attention (AGA) module has been added into the base model as well as adding an ℓ_2 term to the cost function, as shown in Fig. 3.4. The overall structure of the model is depicted in Fig. 3.3.

The model mainly consists of two parts, an encoder which extracts features from the input RGB image at different spatial resolutions, and a decoder which reconstructs the depth map from the features extracted by the encoder. In addition, the encoder and the decoder are connected to each other using an Astrous Spatial Pyramid Pooling (ASPP) module [40] to increase the receptive field of the entire model. All upsampling operations in the model are based on the bilinear resizing method. The whole encoder-decoder structure in the base model [347], itself had been borrowed from [181]. The decoder in [181, 347]

comprises of several adaptive merging blocks (AMB) to fuse features from different levels of the encoder and the decoder, and dilated residual blocks (DRB) modules to increase the receptive field of the encoder and transform the encoder features. AMB blocks, in [181, 347], merge the encoder's features into the decoder's features adaptively which can be considered a channel-wise attention mechanism. The operations in the AMB are nothing but concatenation of both the encoder and the decoder features, followed by the squeeze and excitation operations using the squeeze and excitation networks (SE Networks) [122].

Instead of the AMB block, the improved AGA module as shown in Figs. 3.3 and 3.4 has been added into its most general form. The AGA block benefits from both spatial and channel-wise attention integrated into one module. The first row of operations in Fig. 3.4 is in fact from the AMB module. The novel part of the module is the spatial attention operations which are mixed with the channel-wise operation in an additive and multiplicative fashion. For the spatial attention, the module uses the local cross-correlation of the encoder and the decoder features at each spatial points to shape the encoder features spatially.

At first,theAGA module uses 1×1 convolutions as a bottleneck to go from hyper space (feature space) to embedded Euclidean space for both the encoder and the decoder features. Then the module uses cross-correlation of the embedded features from the encoder and the decoder. More precisely, the module uses absolute value of cosine similarity of the embedded features of the encoder and the decoder at each spatial point as a measure of structural similarity between the depth map features and the RGB features. Since this similarity measure is absolute and normalized, the module can put more constraints on the solution space. As a result, it can shape the RGB features in a better way, both spatially and channel-wise, using the decoder as the representation of the depth features for better depth estimation. See Fig. 3.12 for visual effect of the spatial attention. The operations in the second row and the third row of Fig. 3.4 which calculate the spatial attention (attention maps SA_1 and SA_2) are equivalent to

$$SA_i = \left| cossim\left(E_{l,i}, E_{h,i}\right)\right|, \; i = 1, 2 \tag{3.1}$$

where $E_{l,i}$ and $E_{h,i}$ denote the embedded features of low level features, i.e., F_l or the encoder features, and high level features, i.e., F_h or the decoder features, respectively. The operations in Fig. 3.4 are depicted in this way to facilitate the comprehension of their extension to the non-local AGA module in Fig. 3.8 which will be discussed in Sec. 3.4.5.

Not only the channel-wise attention and the spatial attention are different in the above-mentioned implementation details, but also the purposes of the two are different. The channel-wise attention provides the encoder feature with one scalar multiplicative weight for the entire of each single channel of size $H \times W$. So for the entire encoder's $H \times W \times C$ feature it provides a vector of size $1 \times C$. The vector is scaled before added to the decoder's feature. Spatial attention, instead, is an $H \times W$ attention map that each feature vector at each spatial point of the encoder feature will be multiplied by the corresponding spatial value of the attention map. See Fig. 3.5. The AGA module uses the sensitivity-enhanced absolute value of the above-mentioned cosine similarity. The absolute value enforces the correlation between two feature vectors at each spatial point of the encoder and the decoder features independent of the direction. That is, it compares the presence of any spatial cross-correlation between the depth map features and the RGB features. If there is a correlation between the depth map features and the RGB features, then they carry information about each other regardless of the sign of the correlation. The AGA module in the most general form has been depicted in Fig. 3.4. To go from hyperspace, C, to the embedded space, C_0, at each spatial point, a 1×1 convolution with bottle-necking $C_0 < C$ has been utilized. In this way, the model is able to avoid permutations of the information in different channels since the 2D convolution operation is fully connected in channel direction of the input and summation is permutation indifferent. This bottleneck will give us the structure of the features in that spatial point in the embedded space. This operation is local. Being local and bottle-necked, it is light. The output of this operation is an $H \times W$ spatial attention map. As shown in Fig. 3.4, the AGA module's output is an $H \times W \times C$ tensor of features

$$F_{out} = [f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}] \times F_l + F_h. \tag{3.2}$$

where $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{A}$ stand for spatial, channel-wise and attention, respectively. F_l and F_h are low-level and high-level features from the encoder and the previous stage of the decoder, respectively. The first spatial attention map, $\mathcal{SA}_1$, is additive while the second one, $\mathcal{SA}_2$, is multiplied by the channel-wise attention weights. Element-wise summation and multiplication of tensors of sizes $H \times W \times 1$ and $1 \times 1 \times C$ and $H \times W \times C$ are possible since these operations broadcast the operand tensors. $f_1(\cdot)$ and $f_2(\cdot)$ are introduced to enhance the sensitivity to any non-zero correlation between the high-level features and low-level features in each spatial point. They are chosen either of

- $f(\mathcal{SA}) = \mathcal{SA}$
- $f(\mathcal{SA}) = \mathcal{SA}\ \exp(\mathcal{SA})$

The first one means spatial attention without enhancing sensitivity. The second one means spatial attention with enhanced sensitivity. See Fig. 3.6 for comparison between them. It was experienced that enhancing the sensitivity around 1 helps. One explanation is that the gradients in a normalized output are suppressed. To completely turn off the sensitivity to the additive spatial attention and multiplicative spatial attention, $f_1(\mathcal{SA}) = 0$ and $f_2(\mathcal{SA}) = 1$ are utilized. Our AGA module merges the attended low-level features from each level of the encoder to the corresponding decoder's high-level features. The AGA module will learn the merging parameters, during optimization, to merge the information for all elements of the $H \times W \times C$ encoder tensors weights, i.e., $[f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}]$.

3.3.2 Loss functions

A 3-term cost function is utilized. The virtual normal loss and the weighted cross-entropy loss were already used in the base model [347]. A third term ℓ_2 has been added, based on the $\mathbf{L}_2$ norm of the error. **Virtual Normal Loss (VNL).** The surface normal is an important local feature for 3D reconstruction and depth estimation. However, calculating surface normals in a small area is prone to noise. To remove the effect of noise, [347] suggests to calculate the normals of virtual surfaces built by triangles which their constructing points

have been chosen far from each other in 3D scene at random. If $\mathbf{n}^i_{pred}$ and $\mathbf{n}^i_{gt}$ are the predicted normal and ground truth normal at the point i respectively, then the computed Virtual Normal loss is:

$$\ell_{VN} = \frac{1}{N}\left(\sum_{i=1}^{N} \left\|\mathbf{n}^i_{pred} - \mathbf{n}^i_{gt}\right\|_1\right) \tag{3.3}$$

where N is total number of valid sampled triangles. See [347] for details. Similar results can be achieved if one uses the virtual slope in 3D scene instead of virtual normals. See ablation study in [347]. ℓ_{VN} helps with relative pixel-wise depth values of the predicted depth map and its structure in regression fashion.

Pixel-wise Absolute Depth Supervision. In addition to VNL, there are two terms which enforce pixel-wise absolute depth supervision. Similar to [76, 347], quantized real-valued depth is utilized. So the depth prediction has been formulated as a classification problem instead of regression by employing the cross entropy loss. More precisely, the weighted cross-entropy loss (WCEL) from [33, 347] borrowed, with the weight being the information gain. See [33] for details. Combination of these two above-mentioned terms were already utilized in the based model [347]. In addition to these two terms, the $\mathbf{L}_2$ norm of the difference between the ground truth and the predicted depth map is used, to decrease the root mean square error (RMSE) of the predicted depth map. The WCEL and the VNL and $\mathbf{L}_2$ are combined together to gain an overall supervision in the training phase. So the total loss is

$$\ell = \ell_{WCEL} + \lambda \ell_{VN} + \gamma \ell_2 \tag{3.4}$$

where the weights λ and γ define the contribution of each term. λ has been set to 6 and γ has been set to 25 based on extensive empirical studies. The overall training pipeline is illustrated in Fig. 3.7.

3.4 Experiments and results

The experiments are performed on the NYUDV2 dataset [225] and the KITTI dataset [86] to evaluate the performance of the proposed algorithm in comparison with state-of-the-art

methods. The ablation studies are also performed to better understand the contribution of the different settings of the attention module.

3.4.1 Datasets

NYUDV2. The NYUDV2 dataset consists of 464 different indoor scenes, which are divided into 249 scenes for training and 215 for testing. Similar to [76], the training scenes are used after synchronization using the tool provided by [225] to train the model for the main results and ablation study on NYUDV2. This dataset is referred to as the large NYUDV2. Moreover, a subset of the Raw NYUDV2 dataset has been used which is split to 249/215 train/test split scenes for the ablation study. This dataset is referred to by the small NYUDV2.

KITTI. The KITTI dataset contains over 93K outdoor images and depth maps with an approximate resolution of 1240×374. All images are captured on driving cars by stereo cameras and a Lidar. The test is done on 697 images from 29 scenes split by Eigen et al. [67]. All the images from the scenes in which one of them is in the test scenes are removed and the remaining RGB images and the corresponding ground truth are used in training the model.

3.4.2 Implementation details

Similar to [347], the ResNeXt-101(32 × 4d) [334] encoder pre-trained on ImageNet [58] is used as the encoder in the model. The base model is exactly as described in [347] but all the AMB modules are replaced with the AGA modules. See Fig. 3.3. In the main results (Sec. 3.4.4), the AGA module as described in Sec. 3.3.1 with the additional ℓ_2 loss term are used. All 1×1 bottle-necks in the AGA modules are $\frac{1}{16}$ times of their input channel size.

In all of the experiments the base learning rate is 0.003 used along with a learning rate scheduling going from 1 to 0 linearly for all training procedures on the large NYUDV2 and KITTI and the same learning rate scheduling with power 0.9, is chosen on the small NYUDV2. Stochastic gradient descent is applied as the optimization method with a batch

size of 16 on the large NYUDV2 and the KITTII and a batch size of 8 on the small NYUDV2. The weight decay and momentum are set to 0.0005 and 0.9 respectively. The model is trained for 99300 iterations on large NYUDV2 and KITTI and 5000 iterations on small NYUDV2.

The data augmentation are conducted on the training samples using the following methods. On small and large NYUDV2 the RGB image and the corresponding depth map are randomly resized with ratio [1, 0.92, 0.86, 0.8, 0.75, 0.7, 0.67], randomly flipped horizontally, and finally randomly cropped to 384×384. A similar process is applied for KITTI but resizing with the ratio [1, 1.1, 1.2, 1.3, 1.4, 1.5] and cropping with 384 × 512. Note that the depth map should be scaled to the corresponding resizing ratio [67].

It is worth mentioning that the overall model is similar to what has been used in [347] except the AGA module in magenta in Fig. 3.3. The base model in [347] has exactly 90436054 parameters and only 28672 parameters are added, as a result of adding the AGA module to it, which is around 0.03% (or 0.0003) of the total parameters of the base model. In addition, all added operations are light since they are local.

3.4.3 Evaluation metrics

Similar to [162] the performance of the depth predictions are evaluated quantitatively based on mean absolute relative error (AbsRel), mean log 10 error (log10), root mean squared error (RMS), root mean squared log of error (RMSlog) and the accuracy under threshold ($\sigma_i < 1.25^i\ ,\ i = 1, 2, 3$). See section 1.5 for detailed formula of each measure.

3.4.4 Comparison with state-of-the-art

A comparison of the results with state-of-the-art methods is shown in table 3.2 for large NYUDV2 and in Table 3.3 for the KITTI dataset. As shown in Table 3.2, the suggested method achieves best or comparable results in all the measures except one among all state-of-the-art methods. Examples of the trained model's outputs, largest attention map, ground truth depth map and RGB images are depicted in Fig. 3.12. The attention map shows

stronger response around geometric boundary of the 3D scene as expected by the method. This attention is the strongest at the occlusion boundaries which is an important subset of geometric boundaries. The clear separation of the objects with emphasized geometric boundaries around them suggests that the AGA module is performing as expected by reducing the effect of texture edges and focusing on geometric ones.

The performance on the KITTI datset in comparison with state-of-the-art shows that the methodology is effective on KITTI dataset as well. As it is shown in Table 3.3 the model outperforms the base model [347] in all measures and shows comparable results in comparison with the state-of-the-art in all other measures.

3.4.5 Ablation study

In this section, two sets of experiments are conducted.

Effectiveness of the proposed AGA module over the base model [347] and added ℓ_2 loss term

The effect of different internal settings for the suggested general AGA module depicted in Fig. 3.4 as well as the added ℓ_2 term in the total cost function are examined here. The settings which are referred to by first column of Table 3.4 in this section are the settings for general coefficients of low-level features, i.e., $[f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}]$, in Eq. (3.2). The first row in Table 3.4 is the base model[347] with its cost functions, i.e., VNL and WCEL. Other than the base model and its cost functions in [347] ℓ_2 has been added to the total cost functions. So the different settings with and without ℓ_2 to study the effect of the term are provided here. As Table 3.4 suggests, the structure with first spatial attention with sensitivity enhanced added to the channel attention works the best for the NYUDV2 dataset. However, it is possible that on other data distributions setting $S5$ might be an option because it was noticed that setting $S5$ has less spikes during training in the experiments which is a desirable trait. The root mean square measure (rms) in Table 3.4 is lower for the settings

with the added term ℓ_2 in the total cost function, i.e., $S7$ (lowest) and $S4$ (second lowest). This shows the effectiveness of ℓ_2 term in the total cost function. Note that $S6$ and $S7$ are the same in their AGA setting but the later has the added ℓ_2 term in the total cost function.

Effectiveness of the proposed methodology over the conventional attention

Second, the novelty of the AGA module's implementation is shown, (i.e. sensitivity enhanced absolute value of cosine similarity of the features in embedded space) as a measure of similarity between the encoder's and the decoder's features at each spatial point in comparison with the traditional attention mechanisms. The second set of experiments aims at showing the effectiveness of the above-mentioned cross-correlation measure between the low-level features (the encoder features) as representation of the RGB image and the high-level features (the decoder features) as the representation of the depth map by comparing it to the conventional attention techniques, i.e., dot product and matrix multiplication (non-local operations). Three settings, $DS7$, $NS7$ and $S7$ are compared here. The $S7$ setting has been described in Table 3.4 which is the same for Tables 3.2 and 3.3 as well. The $DS7$ setting is the same as $S7$ but using dot product as coefficients for spatial attention mechanism instead of formula (3.1) and (3.2). $NS7$ is the extension of $S7$ to non-local operations. It compares each spatial points with all points in all other spatial points. The details of implementation of non-local AGA module has been depicted in Fig. 3.8. It is important to note that all three models have exactly the same number of parameters. The only difference is whether the formula (3.1) and (3.2) are used or the attention is local or non-local.

As Table 3.5 suggests, the local AGA module works the best in comparison with the conventional attention mechanisms. The reason that non-local AGA is showing inferior performance in comparison with the suggested (local) AGA module is the introduction of lots of multiplication in forward pass in non-local AGA module in comparison with the local counterparts in the overall model. Those multiplications create complications in gradients, as a result the optimization process become less efficient. In addition, absolute

value of cosine similarity is normalized and sign indifferent and measures similarity as far as there is a cross-correlation between the two source of information while dot product does not consider these two. In other words, the absolute value of cosine similarity is absolute and normalized version of dot product which means imposing more constraint on the network to regulate the solution space better. Also the sensitivity is enhanced at any non-zero correlation. The matrix multiplication create the non-local version of operations which are computationally costly and not much effective as well.

3.5 Discussion and conclusion

The main idea of [224] was taking advantage of the similarity of the RGB image and the depth map in the area of the 3D scene close to geometric edges. In other words, it is desired to guide the encoder to shape better RGB features using the depth map for better depth estimation at each spatial point. The eventual criteria for this guidance is the sensitivity enhanced absolute value of cosine similarity between the local embedded features at every spatial point of the encoder and the decoder. It is allowed to be done since the features in the decoder are close to the end of the model and closer to the cost function in training phase.

The benefits of using absolute value of local cosine similarity in embedded space in comparison with the conventional attention techniques, i.e., dot product, is that it is absolute and normalized so it puts stronger constraints on the network to regulate solution space better. It is also local so it does not create difficulty in optimization with matrix multiplications in non-local versions. It is important to note that for designing the suggested AGA module which uses the guidance of the depth map features to shape the RGB features, one might be able to assign more time and hardware resources to find more effective complex operations instead of $f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}$ in Fig. 3.4 and (3.2). However, fine tuning the structure and parameters of such a module would be difficult. Hence, it was decided to use the divide-and-conquer strategy, where the guidance is divided into

additive and multiplicative spatial attention weights, $f_1(\mathcal{SA}_1)$ and $f_2(\mathcal{SA}_2)$, and channel-wise attention weights $\mathcal{CA}$.

At the end it is worse mentioning that this research was subsection of a larger research which was aiming at autonomous navigation of indoor flying robots. For navigation, absolute relative error is the most important measure along with accuracy of prediction $\sigma_i,\ i = 1, 2, 3$.

There are some other directions that has been explored.

3.5.1 Using principal component analysis (PCA)

Principal component analysis is a strong tool for model reduction. As a result the following steps were tried:

- reshape depth images to vectors
- calculate the eigen values and eigen vectors of the depth images in NYUDv2 datset.
- keep the most important eigen vectors
- extract the coefficient of those eigen vectors from corresponding RGB images using deep-learning.
- filter the outputs using deep learning at the output size

The best AbsRel error was around 12%. Comparing the results in Tab 3.2, which was around 9.7%, with the result of this methods shows the idea is not working. The reason is that the dataset is not large enough to train the network in this way.

3.5.2 Utilization of spatial regularization effect in dense depth estimation

In traditional automatic feature extraction methods in CNNs, one usually reduces spacial size by factor of 2 and increase the number of feature layers by factor of 2 which will result

in reduction of total features by factor of 2 ($\frac{1}{2} \times \frac{1}{2} \times 2 = \frac{1}{2}$). This practice is done in classification algorithms mainly where one feature vector is extracted for the entire image to describe the whole scene. However, this would push most weights to be in smaller spacial sizes which in turn results in less generalization power for dense feature extraction tasks like depth estimation.

On the other hand, in dense feature extraction for depth estimation, it is necessary to achieve non-local perception of the scene at each spatial point. The convolutions' main benefit is sharing the weights between different spatial points in the feature tensor to achieve better generalization power. While fully connected layers are too flexible and that is the reason why they generalize poorly in comparison whith CNNs. So the question becomes why one does not use more spatial size for better generalization. In the following, we describe some preliminary works we conducted along this direction. The section 3.5.2 talks about this research idea. This idea is applicable in all tasks which a dense feature extraction, like depth estimation, is involved. **Extending DRB to build an encoder-decoder structure** Let's take a look at the internal structure of the DRB block in Fig. 3.3. See Fig. 3.9. The structure of the DRB block has been extended to Fig. 3.10 to be used in an encoder-decoder structure depicted in Fig. 3.11. The AGA modules have been adapted to this new structure so that it can get attention from unequal features in channel direction. Cost-functions have been borrowed from [224]. The results of training this network on KITTI dataset is available in Table 3.1. It is worth mentioning that the network has been trained just on NYUDv2 while model in [224] was trained on ImageNet first and the NYUDv2. So it is not fair to the algorithm in this section to be compared with the algorithm in [224]. The extra ImageNet help the model to learn more robust features at least in their encoder. In addition, using some methods like ResNet and ResNext one can reduce the number of parameters that will help with generalization. The last point is that doing a pruning algorithm makes sense as the network has lots of parameters.

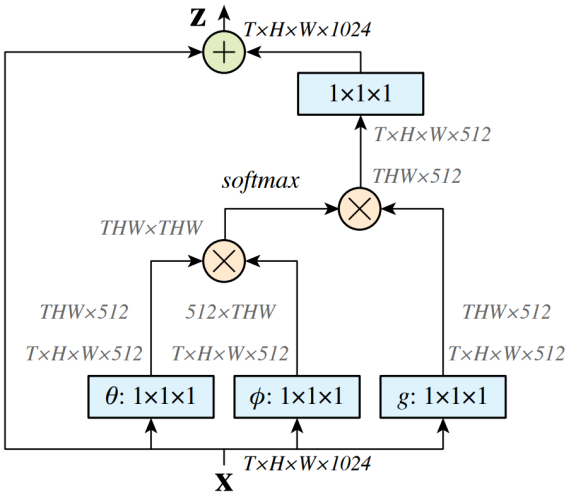


Figure 3.1: Comon practice for available attention mechanisms and transformers which are based on attention mechanisms which is a space-time non-local block [322]. The feature maps are shown as the shape of their tensors, e.g., $T \times H \times W \times 1024$ for 1024 channels (proper reshaping is performed when noted). "$\otimes$" denotes matrix multiplication, and "$\oplus$" denotes element-wise sum. The softmax operation is performed on each row. The blue boxes denote $1 \times 1 \times 1$ convolutions. This figure shows the embedded Gaussian version, with a bottleneck of 512 channels. The vanilla Gaussian version can be done by removing θ and ϕ , and the dot-product version can be done by replacing softmax with scaling by $1/N$. The figure is from [322].

Figure 3.2: Comparison of edges and gradients in an RGB image and the corresponding depth map. Top-left: RGB image. Top-right: the corresponding depth map of the RGB image. Bottom-left: Laplacian of the RGB image. Bottom-right: Laplacian of the depth map. Figure from [224].

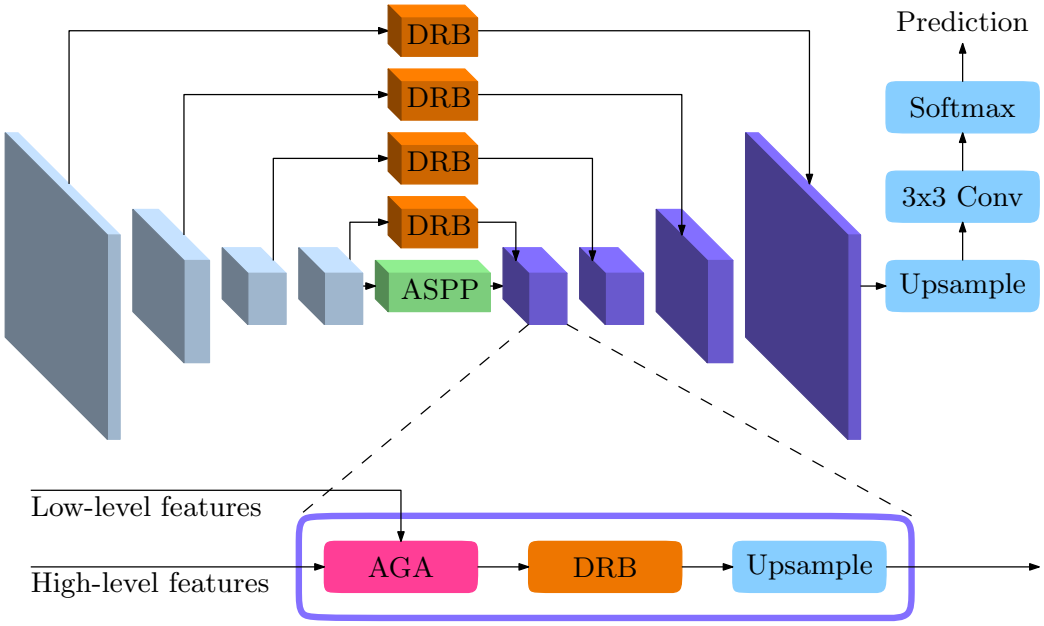


Figure 3.3: An overall structure of the model. Figure from [224].

Table 3.1: Results on KITTI dataset as compared with state-of-the-art methods. The best result in each column (measure) is depicted in bold text. The second best is underlined.the new model (**EDRB**) based on Extended DRB and modified AGA module shows comparable performance with other state state-of-the-art methods. See section 1.5 for detailed formula of each measure.

Method	Err(lower is better)			Acc(higher is better)		
	AbsRel	RMSE	RMSElog	σ_1	σ_2	σ_3
Su [283]	0.117	4.251	0.174	0.894	0.971	0.984
Fang [70]	0.098	4.075	0.174	0.889	0.963	0.985
Wang [314]	0.096	4.327	0.171	0.893	0.963	0.983
EDRB	0.073	3.327	0.117	0.940	0.990	**0.998**
Fu [76]	0.072	**2.727**	0.120	0.932	0.984	0.994
Liu [197]	<u>0.070</u>	2.912	0.121	0.942	0.986	0.992
Lee [168]	**0.059**	<u>2.756</u>	**0.096**	**0.956**	**0.993**	**0.998**
base [347]	0.072	3.258	0.117	0.938	0.990	**0.998**
ours	<u>0.070</u>	3.223	<u>0.113</u>	<u>0.944</u>	<u>0.991</u>	**0.998**

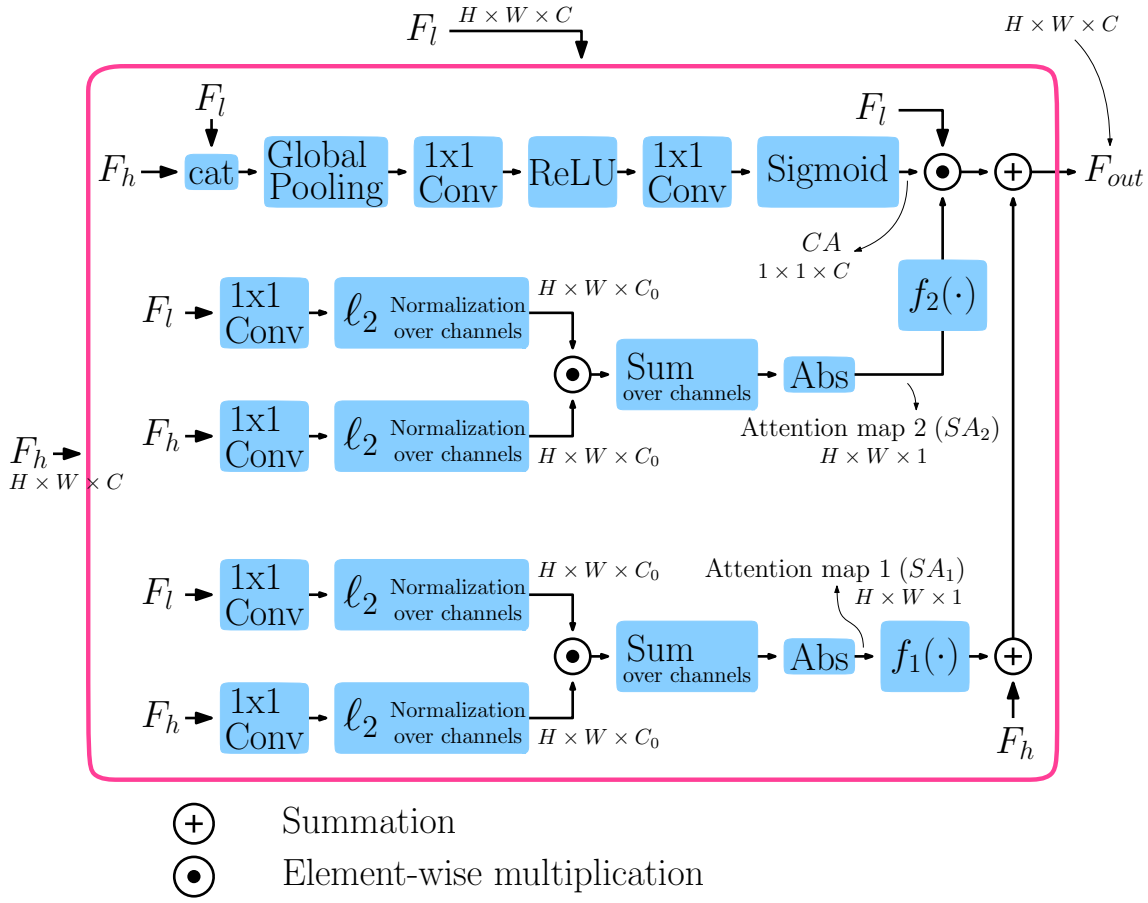


Figure 3.4: The internal structure of AGA module in its most general settings. $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{A}$ stand for spatial, channel-wise and attention, respectively. F_l and F_h are low-level and high-level features from the encoder and the previous stage of the decoder, respectively. The attention maps SA_1 and SA_2 are discussed in Sec. 3.3.1 and are equivalent to (3.1). Figure from [224].

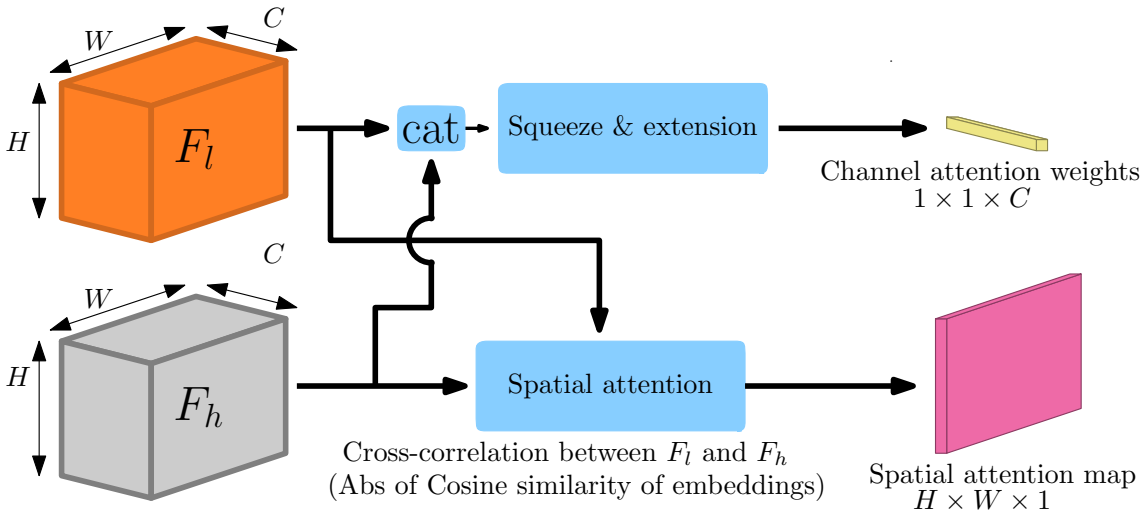


Figure 3.5: Illustration of the differences between the spatial attention and the channel-wise attention discussed in Sec. 3.3.1. Figure from [224].

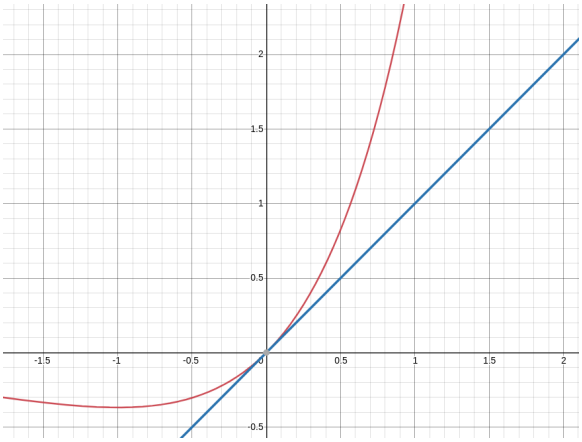


Figure 3.6: Comparing $x \exp(x)$, in red color, and x, in blue color. Figure from [224].

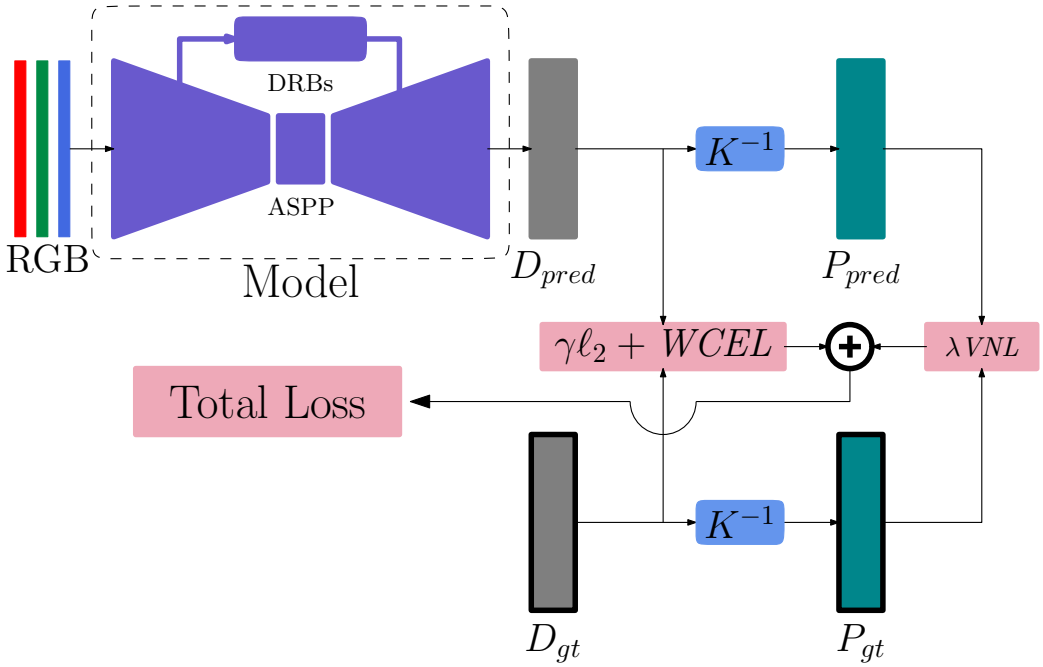


Figure 3.7: The overall training pipeline. Total loss consists of three terms ℓ_{WCEL}, ℓ_{VN} and ℓ_2. ℓ_{WCEL} and ℓ_2 compare the absolute predicted depth map, D_{pred}, and the ground truth depth map, D_{gt}. ℓ_{VN} compares the virtual normals using the predicted point cloud, P_{pred}, and the ground truth point cloud P_{gt}. γ and λ are scaling constants tuned to give an appropriate effect to each term in the total cost function. ℓ_{WCEL}, ℓ_{VN} are from the base model [347]. Figure from [224].

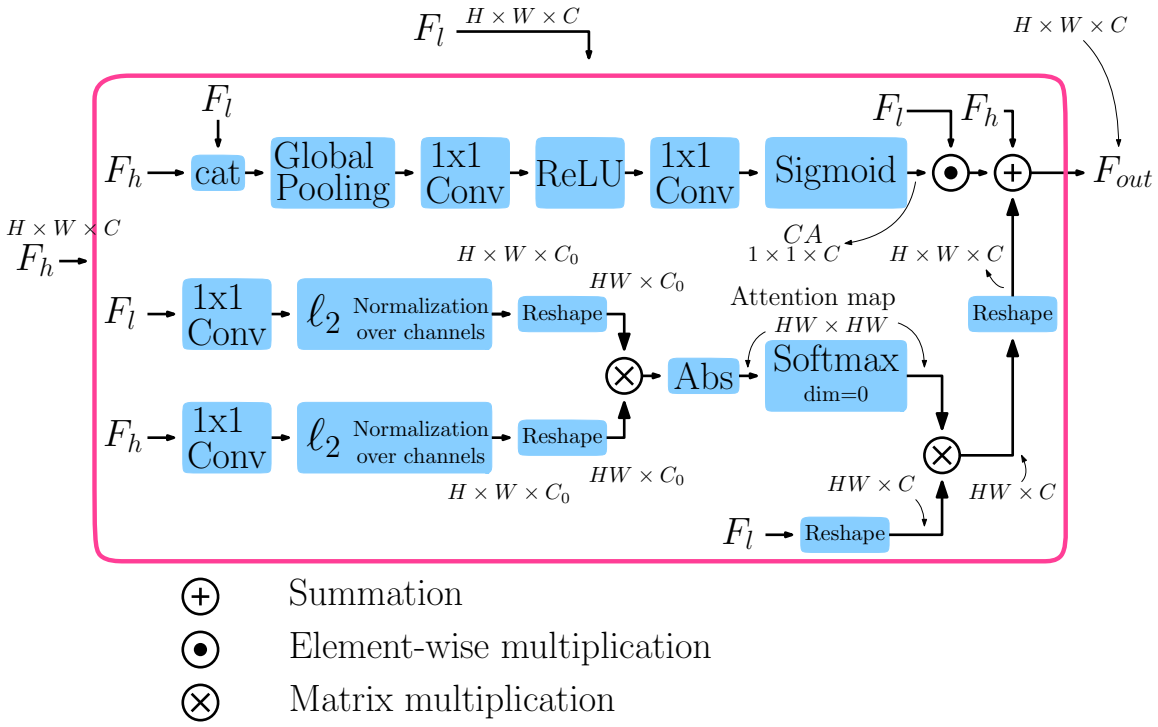


Figure 3.8: The internal structure of the Non-local AGA module as the natural extension from the setting for the AGA modules for the main results of the paper [224]. Figure from [224].

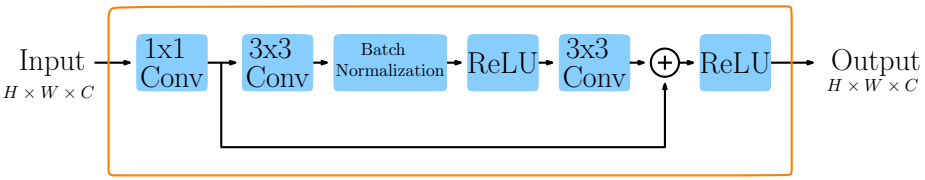


Figure 3.9: Internal structure of the DRB block in [224]

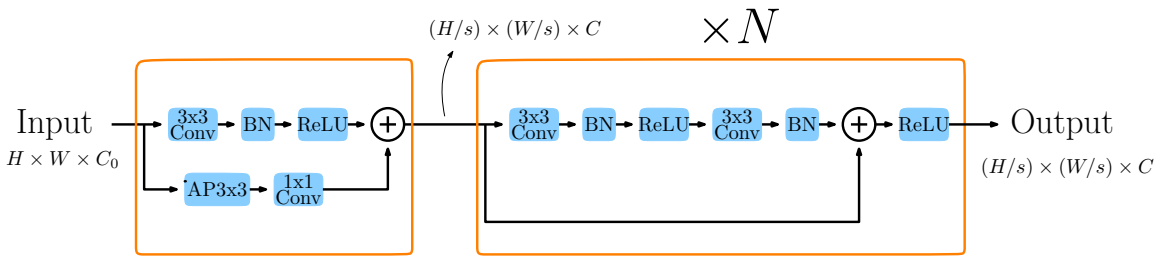


Figure 3.10: Internal structure of the extended DRB block. The second part is repeated N times. If the stride, s, of the first convolution in the first block is equal to 1 then the average pool won't be needed.

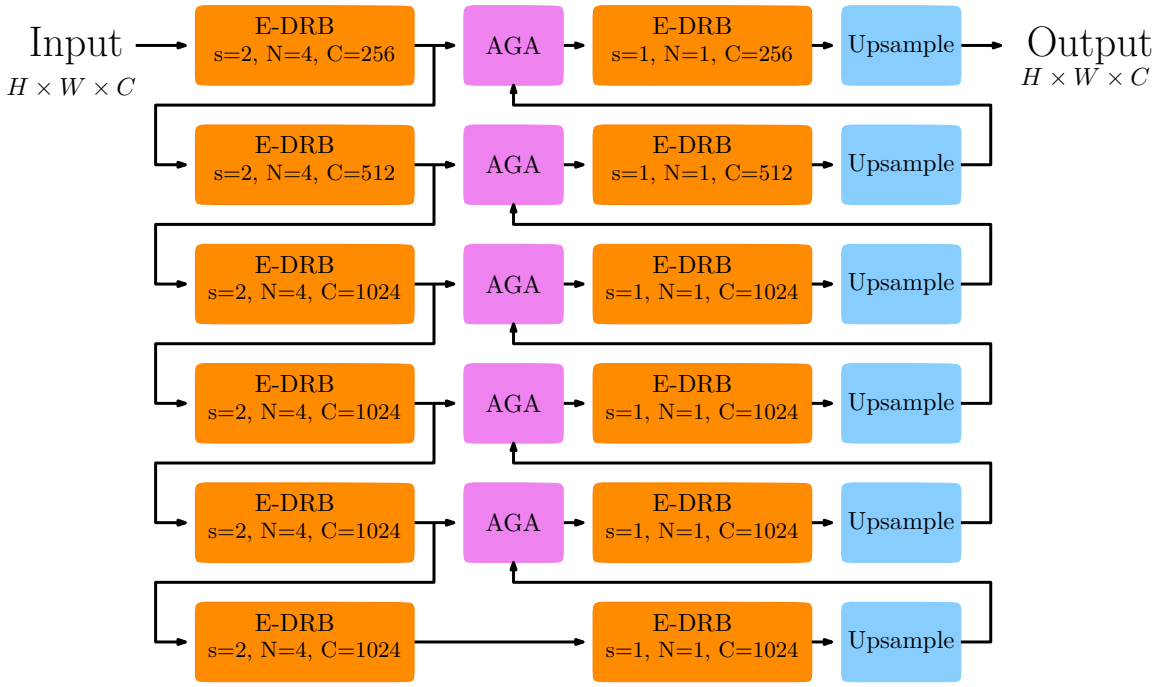


Figure 3.11: Suggested encoder-decoder structure based on Extended-DRB and AGA modules. The AGA modules has been adapted to this new structure so that it can get attention from inequal features in channel direction. Upsampling has been done using bilinear upsampling just like [224].

Table 3.2: Results on large NYUDV2 as compared to other state-of-the-art methods. The best result in each column (measure) is depicted in bold text. The second best is underlined. Table from [224]. See section 1.5 for detailed formula of each measure.

Method	Err(lower is better)			Acc(higher is better)		
	AbsRel	log10	RMS	σ_1	σ_2	σ_3
Ladicky [161]	-	-	-	0.542	0.829	0.941
Liu [196]	0.327	0.126	1.08	-	-	-
Zhuo [362]	0.305	0.122	1.04	0.525	0.838	0.962
Liu [190]	0.230	0.095	0.824	0.614	0.883	0.971
Li [176]	0.223	0.091	0.759	0.640	0.900	0.974
Wang [319]	0.220	0.094	0.745	0.605	0.890	0.970
Eigen [67]	0.215	-	0.907	0.611	0.887	0.971
Eigen [66]	0.158	-	0.641	0.769	0.950	0.988
Chakrab [36]	0.149	-	0.620	0.806	0.958	0.987
Li [179]	0.143	0.063	0.635	0.788	0.958	0.991
Su [283]	0.137	0.058	0.498	0.826	0.967	0.995
Qi [245]	0.128	0.057	0.569	0.834	0.960	0.990
Wang[316]	0.128	-	0.497	0.845	0.966	0.990
Wang [317]	0.128	-	0.493	0.844	0.964	0.991
Laina [162]	0.127	0.055	0.573	0.811	0.953	0.988
Xu [335]	0.121	0.052	0.586	0.811	0.954	0.987
Lee [167]	0.119	0.050	0.430	0.870	0.974	0.993
Wang [314]	0.115	0.049	0.519	0.871	0.975	0.993
Fu [76]	0.115	0.051	0.509	0.828	0.965	0.992
Hu [123]	0.115	0.050	0.530	0.866	0.975	0.993
Liu [197]	0.113	0.049	0.523	0.872	0.975	0.993
Lee [168]	0.110	0.047	<u>0.392</u>	0.885	0.978	0.994
Huynh [127]	0.108	-	0.412	0.882	0.980	**0.996**
Fang [70]	0.101	-	0.412	0.868	0.958	0.986
Yang [344]	0.106	0.045	**0.365**	**0.900**	**0.983**	**0.996**
base [347]	0.108	0.048	0.416	0.875	0.976	0.994
ours	**0.097**	**0.042**	0.444	<u>0.897</u>	<u>0.982</u>	**0.996**

Table 3.3: Results on KITTI dataset as compared with state-of-the-art methods. The best result in each column (measure) is depicted in bold text. The second best is underlined. the model consistently beats the base model [347] in all measures and shows comparable performance with other state state-of-the-art methods. Table from [224]. See section 1.5 for detailed formula of each measure.

Method	Err(lower is better)			Acc(higher is better)		
	AbsRel	RMS	RMSlog	σ_1	σ_2	σ_3
Su [283]	0.117	4.251	0.174	0.894	0.971	0.984
Fang [70]	0.098	4.075	0.174	0.889	0.963	0.985
Wang [314]	0.096	4.327	0.171	0.893	0.963	0.983
Fu [76]	0.072	**2.727**	0.120	0.932	0.984	0.994
Liu [197]	<u>0.070</u>	2.912	0.121	0.942	0.986	0.992
Lee [168]	**0.059**	<u>2.756</u>	**0.096**	**0.956**	**0.993**	**0.998**
base [347]	0.072	3.258	0.117	0.938	0.990	**0.998**
ours	<u>0.070</u>	3.223	<u>0.113</u>	<u>0.944</u>	<u>0.991</u>	**0.998**

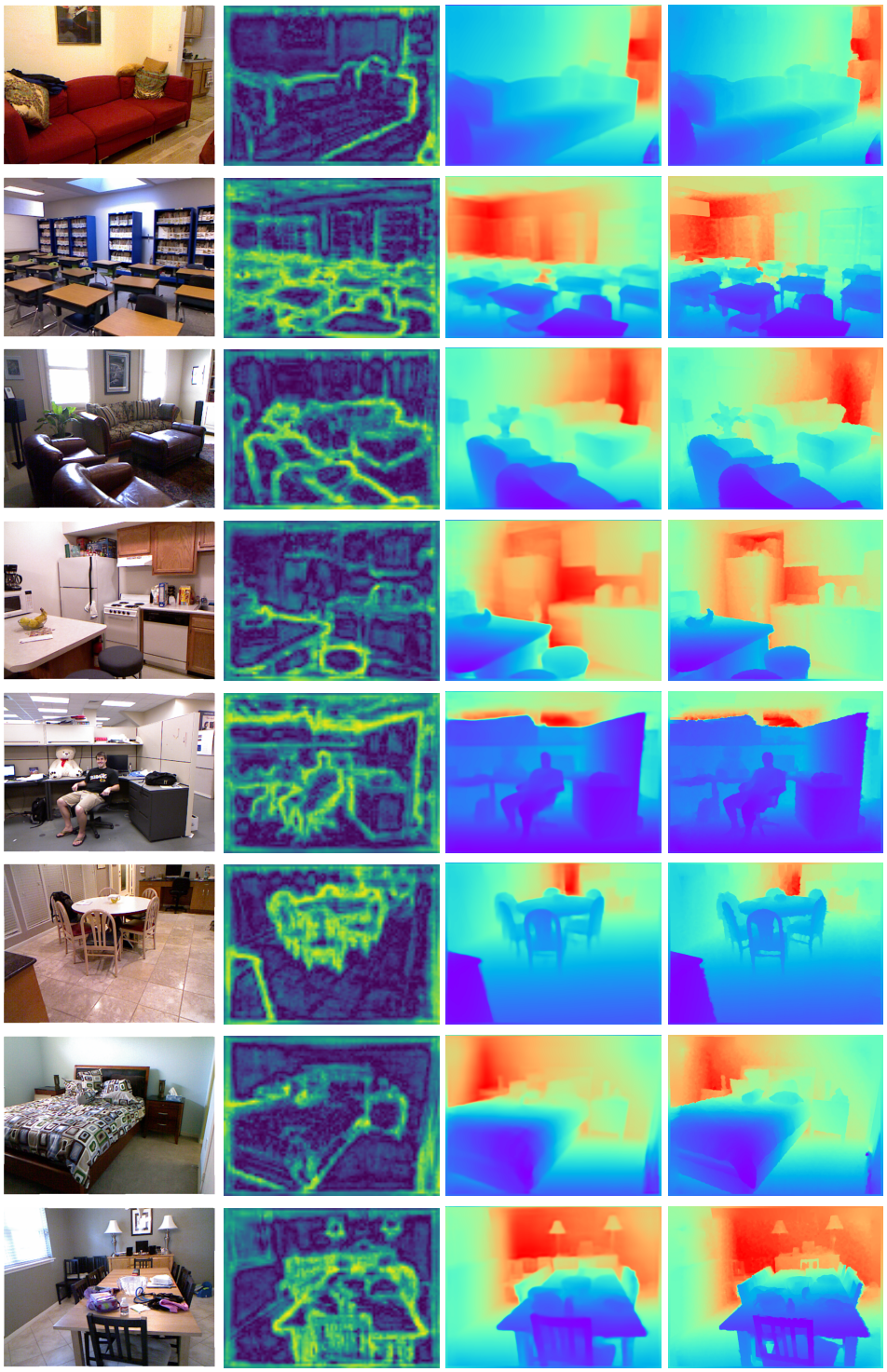


Figure 3.12: Qualitative results. From left to right: RGB image, attention map, predicted depth map, ground truth depth map. As the attention map depicts, the attention is higher at the geometric boundary of the 3D scene. This attention is strongest at occlusion boundaries which is an important subset of geometric boundaries.

Table 3.4: Ablation study for different settings in the suggested general AGA module in Fig. 3.4 compared to the base model [347]. The settings which are referred to by first column of this table are the settings for general coefficients of low-level features, i.e. $[f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}]$, in Eq. (3.2). The results are the last iteration of each experiment which are filtered using a moving average with length 15. In this table the first row represents $\mathcal{CA}$ is the base model [347]. $S2 = \mathcal{SA}_2 \times \mathcal{CA}$. $S3 = \mathcal{SA}_2$. $S4 = \mathcal{SA}_2 \times \mathcal{CA}$ as well as added ℓ_2 in total cost-function. $S5 = \mathcal{SA}_1 \times exp(\mathcal{SA}_1) + \mathcal{SA}_2 \times \mathcal{CA}$. $S6 = \mathcal{SA}_1 \times exp(\mathcal{SA}_1) + \mathcal{CA}$. $S7 = \mathcal{SA}_1 \times exp(\mathcal{SA}_1) + \mathcal{CA}$ as well as added ℓ_2 in total cost-function. $\mathcal{S}$, $\mathcal{C}$ and $\mathcal{A}$ stand for spatial, channel-wise and attention respectively. All settings have been trained with $\ell = \ell_{WCEL} + \lambda\ell_{VN}$, but the ones with added ℓ_2 trained using $\ell = \ell_{WCEL} + \lambda\ell_{VN} + \gamma\ell_2$. The best value in each column is bold type and the second best is underlined. See section 1.5 for detailed formula of each measure. Table from [224].

Set.	Error (lower is better)			Acc (higher is better)		
	AbsRel	log10	RMS	σ_1	σ_2	σ_3
[347]	0.1408	0.0590	0.5951	0.8217	0.9635	0.9907
S2	0.1385	0.0581	0.5856	0.8269	0.9644	0.9915
S3	0.1388	0.0586	0.5980	0.8232	0.9641	0.9912
S4	0.1381	0.0578	<u>0.5702</u>	0.8277	0.9643	0.9919
S5	<u>0.1361</u>	0.0577	0.5832	0.8283	0.9658	0.9916
S6	**0.1345**	<u>0.0574</u>	0.5904	<u>0.8302</u>	<u>0.9670</u>	<u>0.9921</u>
S7	0.1364	**0.0568**	**0.5567**	**0.8319**	**0.9671**	**0.9929**

Table 3.5: Comparison between the AGA module and the conventional attention techniques, dot product and costly matrix multiplication. The $S7$ is just like the setting of main results of table 3.2, 3.3 and table 3.5. In this table the first row, $DS7$, is the same setting in $S7$ but using dot product as spactial attention mechanism instead of the similarity measure. $NS7$ is extension of $S7$ to non-local operation and compare each local feature vector with all other points in other spatial points. Table from [224]. See section 1.5 for detailed formula of each measure.

Set.	Error (lower is better)			Acc (higher is better)		
	AbsRel	log10	RMS	σ_1	σ_2	σ_3
DS7	0.102	0.045	0.452	0.881	0.974	0.994
NS7	0.101	0.045	0.450	0.882	0.976	0.994
S7	**0.097**	**0.042**	**0.444**	**0.897**	**0.982**	**0.996**

Chapter 4

MDE in Dynamic Scene, Literature Survey

Although results in SfM and visual SLAM are impressive, most algorithms assume the scene is static. However, the real world scene consists of dynamic objects. This discrepancy results in erroneous estimations [284]. So it is necessary to do estimation of both mapping and localization in a robust way. There are lots of applications for such a scheme like robot navigation [217], [24], autonomous driving systems in automobiles [213], [274], or emergency response missions [49], [249]. Not only reconstruction of the scene but also to some extend capability of the algorithm in detection, the shape of the dynamic objects and tracking them accurately play an important role in autonomous cars and navigation of robots.

With this aim in mind, [311] and [310] utilized a Bayesian approach on the outputs of tracking which are themselves outputs of a laser scanner. They aim at tracking moving objects and the system is called simultaneous localization, mapping, and moving object tracking or SLMMOT. Multibody structure from motion algorithm or MBSfM is the natural generalization of the SfM algorithm to rigid multi-motion models was studied in the computer vision community [25], [52]. Since mobile and wearable devices are available

everywhere, the MBSfM in dynamic scene is beneficial in many downstream applications, such as human-robot interaction [95], obstacle avoidance [120], people-following drones [260], cooperative robotics [88], path planning [38], collaborative mapping [364], driverless cars [213], augmented reality such as cell phones [148], devices which are wearable [35], and assistance for visually impaired individuals in navigation [7], [261].

4.1 Classification of existing approaches

Depth estimation, 3D reconstruction, SLAM and SfM in dynamic scene can be addressed in two different ways. It can be solved as a robustness problem or it can be explicitly modeled which would be an extended version of the standard multi-view geometry model. The first methodology is completely possible under the condition that there are not many moving objects in the scene or the scene is not congested or there is not a large moving object in front of camera. Other than that the robust solution might fail. These robust algorithms are possible if one is able to segment the scene into static as foreground and dynamic objects as background and then ignore the back ground or the dynamic objects. However, the structure of the scene at the moving objects would not be accurate even if the localization is robust.

The extension to the static scene multi-view geometry method should segment the scene into different clusters and associate them with different objects in the scene. Then, objects can be reconstructed and their path can be tracked. However, combining all different motion models is a challenge due to the inherent scale ambiguity in MDE problem.

Existing methods can be categorized into

A. Casting the problem as purely a robustness problem and ignore the dynamic behavior.

B. Casting the problem as a segmentation of the objects in the scene based on their dynamics (their rotation and their translation) and then Tracking them in 3D

C. Casting the problem as simultaneous motion segmentation and reconstruction

See Fig. 4.1 for general structure of Visual Slam/SfM in dynamic environment.

4.2 Robust MDE

This category of algorithms comprise of two parts. First, motion segmentation, second, ego-motion estimation and 3D mapping. The first part divides the scene into the dynamic segments (the background in this case) and the static segment (which is the foreground in this case). Then it utilizes just the static segments for the estimation of the scene. The results of this sub algorithm can be used in algorithms in module B for further processing. See Fig. 4.2.

4.2.1 Motion segmentation

Motion segmentation [59], [146] and [159] is an algorithm that distinguishes dynamic parts of the scene. Standard SLAM/SfM does that by utilizing robust statistical methods like Random Sample Consensus (RANSAC) [72]. The algorithm does not consider the points in the scene which results in high error in geometric model. [112] utilizes the Sampson distance to find the points to exclude.

However, this approach is possible if the scene is not congested by dynamic objects or there is no big moving object in front of the camera. In this case other methods should be utilized. For example external sensors like IMUs can fix the issue [137], [174]. It is done since fusing the IMU signal is lead to improve the localization as well as the segmentation of the scene in accuracy.

Methods which initialize the scene into the foreground and the background

These methods benefits from some knowledge about the scene to segment the scene into the dynamic sections and the static section.

Most algorithms in foreground initialization exploit a technique which is called tracking by detection [27], [169]. Somkiat Wangsiripitak and David Murray [324] use a polyhedral model which its edge points are tracked utilizing Harris's RaPid tracker [108]. Another similar approach was done in [323]. Chhaya et al. [46] models the cars which are against the camera making use of an object class model which is flexible. Then their model is trained utilizing the Principal Component Analysis (PCA). They segment the scene using the trained model.

Background initialization can be found in background subtraction techniques [16], [237].

Geometric constraints

These methods utilize geometric constraints, i.e. epipolar geometry,[110] to divide the scene into the static and the dynamic segments. These algorithms are possible because the points which belongs to dynamic objects do not conform to the multi-view epipolar geometry. See Figure 4.3.

However, these methods fail when there is a degeneracy in the dynamic points i.e. when they moves along the epipolar line in the 3D scene. Kundu et al. [159] address this issue by constraining the error using Flow Vector Bound (FVB) and then detect the static scene using a recursive Bayes filter.

Static-dynamic segmentation can be done using PnP as well. Migliore et al. [215] does that utilizing triangulation. They consider the intersections using a probabilistic filtering algorithm i.e. Uncertain Projective Geometry [117].

In addition, reconstruction of the RGB images from consecutive frames itself can be exploited as a geometric measure to classify the points into static or dynamic as well as detecting occlusion [364], [284].

Optical flow

Optical flow defines the way the points which belongs to the objects, the surfaces, and the edges in a scene move. This is caused by the motion between the observer relative to the scene [28]. It can also be defined as the distribution of apparent velocities of movement of brightness pattern between two consecutive frames [119]. Generally speaking, these flow express the motion of the objects in a scene. As a result, it can be exploited for motion segmentation task.

Klappstein [146] design a metric computed using the optical flow. Then a moving object likelihood designed based on the metric. The designed metric expresses how much the optical flow is failed to comply with. Then, the segmentation of the moving objects is done utilizing the graph-cut algorithm on the motion metric.

Alcantarilla et al. [7] exploit the scene flow to segment the scene. This is done utilizing the residual motion likelihoods to discriminate the static from the dynamic part of the scene.

Derome et al. [59], [60] first estimate a construction residual using a stereo camera in time. Then the motion segmentation is done by finding anomaly in the residual field.

Kopf et al. [153] present an algorithm to predict the depth maps in a consistent and dense fashion and ego-motion from a monocular video. They integrate a learning-based depth prior, in the form of a CNN trained to predict the depth map in a single-image manner, with geometric optimization, to estimate a smooth camera trajectory as well as detailed and stable depth reconstruction. The algorithm combines two complementary techniques: (1) flexible deformation-splines for low-frequency large-scale alignment and (2) geometry-aware depth filtering for high-frequency alignment of fine depth details.

Constraining ego-motion

The visual SLAM and SfM algorithms estimate the ego-motion using the 5-point method [226] or the 8-point [199] method. The aforementioned algorithms do not have any

assumptions on the type of motion. However, sometimes there exist some physical limitations based on the mechanical setting of the camera motion. Parameterization of the camera motion based on these constraints makes the overall estimation more precise [265, 264, 258].

Deep learning for motion segmentation

Deep Neural Networks became popular in computer vision after their success in the ImageNet object recognition competition [154]. They offer an automatic learning method to represent features automatically. The reason behind their success id that they can gain high-level understanding of the scene by learning high-level features from low level; feature automatically [102], [166]. The DNNs have significantly revolutionized many research areas [105].

It has been well-known that the motion segmentation is achievable using optical flow even if one uses feature-based methods. Dosovitskiy et al. [65] proposed supervised optical flow learning based on CNNs (FlowNetS, FlowNetC). Later on the model was made better by combining the two previous architecture [129] (FlowNet 2.0).

One of the best works in the area of optical flow estimation is RAFT [285]. RAFT exploits both DNNs and RNNs. It has 3 components: First an encoder that extracts dense features from the two RGB images and a context encoder that extracts dense features from only first RGB image. Second a layer which calculates the correlations between the two extracted dense features from RGB images 1 and 2 and outputs a 4D $W \times H \times W \times H$ correlation volume. The correlation volume calculates the inner product of every two pairs of the feature. Then using a RNN to fine-tune the optical flow values of the context encoder in a recurrent fashion by using the 4D correlation volume. See Fig. 4.4. Mayer et al. [210] extend the optical flow to scene flow using stereo pairs. This scene flow can be further processed using DNNs to extract the motion features [90]. These features are useful in other tasks like action recognition [89], [276]. It is not clear whether the algorithm is efficient in motion segmentation though.

A network is built by Lin and Wang [188] to partition dynamic objects in a photo space explicitly. To learn spatio-temporal characteristics, they utilize an algorithm which is called reconstruction independent component analysis autoencoders [163], [164]. However, since the spatio-temporal features are unable to understand the 3D geometry of the moving parts of the scene, the features which are geometric are also utilized to cluster the moving objects in the image space. For the purpose of final motion segmentation, recursive neural networks (RNNs) are fed with geometrical and spatiotemporal data.

Recent research by Valipour et al. [300] suggests using the recurrent fully convolutional network, which is abbreviated as RFCN, to segregate the foreground, i.e. the moving objhects in the image space, in the sequences of frames while incorporating temporal data. A gated recurrent unit is utilized to model temporal information ahead of the deconvolution layers. To learn spatial features, a fully convolutional network [198] is used which provides the dense estimation.

Fragkiadaki et al. [75] segment dynamic parts of the scene utilizing a "objectness score" given the optical flow and the color image, which is a distinct method. To preprocess the optical flow and the color images and for creating the motion proposal, two parallel CNNs that are identical to AlexNet [154] are built.

4.2.2 Localization and three-dimensional reconstruction

The predictions of the ego-motion and the three-dimensional geometric structure of the scene from several frames are referred to as localization and 3D reconstruction. This is accomplished by utilizing features which are matched in conventional visual SLAM. Let p be the total number of points, and assume $\{x_{1j}, x_{2j}\}_{j=1}^{p} \in \mathbb{P}^2$ are the features which are matched between the two consecutive frames 1 and 2. By applying epipolar geometry [110] to the feature correspondences, visual SLAM calculates the camera position, which

includes a 3D vector $t \in \mathbb{R}^3$, which indicates the three dimensional relocation, and $R \in SO(3)$, which indicates a rotation matrix in 3D, as well as the point cloud in 3D $\{X_j\}_{j=1}^{p} \in \mathbb{P}^3$. In robust visual SLAM, only static features are used to compute the R, t and the 3D point cloud. The computation is done without using any dynamic features because they are all considered outliers. Instead of constructing feature correspondences, deep learning approaches can handle the image sequences directly. The approaches in the estimation of the rotation and translation and 3D point clouds, which are based on feature extraction or based on deep learning, are covered in this section.

Feature-based approaches

Salient features are retrieved in feature-based visual SLAM to address the picture correspondence issue. There are numerous feature extraction approaches that have been developed by the computer vision researchers. Recent researches [270], [330] usually employ robust feature detection algorithms like SIFT[200] or its lightweight equivalents SURF[19]. The first research in SfM [292], such as the well-known "Visual Odometry" [227] used the Harris corner detector [109]. For real-time applications, however, a quicker method like Features from Accelerated Segment Test (FAST)[255] is used because SIFT and SURF are considered to be computationally expensive[187], [148].

Feature-matching techniques are used to compare extracted features in order to find correspondences. The baseline/parallax, or separation between the optical centers of two cameras, can be used to categorize the approaches. Short baselines can be matched using optical flow-based methods like the Kanade-Lucas-Tomashi (KLT) tracker [201]. On the other hand, highly discriminative feature descriptors which are stringly discriminative are required to find correspondences in long baselines. It is done by computing the dissimilarity between those descriptors (like BRISK [173], SIFT [200], BRIEF [31], SURF [19], etc.). Nevertheless, there is no way to ensure precise correspondences when utilizing these feature-matching algorithms in cases that outliers are prevalent. Implementing estimators

which are robust to outliers, such as PROSAC [48], PROSAC[48], RANSAC [72], etc., is helpful in handling spurious correspondences and rejecting outliers.

The R, t between 2 or 3 frames can be reconstructed if the image correspondences are known. However, the reconstructed scene does not have the correct scale in MDE. The 8-point[199] or 5-point algorithm[226] can compute the posture from two views when the epipolar constraint is enforced, whereas the tri-focal tensor [291] is proposed to deal with three consecutive frames. By enforcing the perspective-n-point constraints in case that the three dimensional point-clouds of the scene are already estimated, motion model with regard to the 3D structure can be produced (like P3P algorithm [82]).

By enforcing triangulation to intersect two projection ray lines, it is simple to reconstruct the three -dimensional structure of the scene once the camera posture has been obtained. The midpoint approach [20] or least square paradigm [113] is suggested to estimate the intersection since the rays don't always intersect as a result of incorrect correspondences. Bundle adjustment [331] is then utilized to optimize the R, t and the three-dimensional point-clouds via reprojection errors minimization in order to prevent the drifting issue. Levenberg-Marquardt (LM) optimization, a variation of the Gauss-Newton method, is the widely used technique to jointly optimize the scene's structure and camera motion.

There are a few different ways to put feature-based visual SLAM into practice. Mouragnon et al. [218], [219] suggest using local bundle adjustment to improve the last few frames rather than improving the R, t and the three dimensional pointclouds of the surroundings across all images (LBA). "PTAM," developed by Klein and Murray [147], demonstrates how the "tracking" and the "mapping" can proceed real-time when the estimation is carried out by various threads. Additionally, the algorithm utilize the concept of selecting the "key frames"; hence, LBA may be applied on the "key frames".

However, Lim et al. [187] employed a "metric topological mapping" and a "binary

descriptors" to enable large scale mapping to function in a real-time fashion which does not need any parallel processing. Statistical model selection [290], ORB features [256], the loop closures that utilizes the "bag of words", the "place recognition" [54], [79], graph optimization [157], and local bundle adjustment [219] are some of the recent state-of-the-art techniques that merge hardware and algorithmic innovation in the last ten years (like ORB-SLAM [221]). Readers interested in a more thorough analysis of common feature-based methods can refer to [77] or [349].

Application of deep learning in pose and depth map and 3D structure predictions

Recently miscellanies works which utilize deep-learning has been very successful in different prediction problems like NLP and computer vision. As a result of this success, researchers cast R, t prediction as a deep-learning paradigm as well. There are few end-to-end deep-learning systems for three dimensional prediction of the structure of the scene, despite the fact that there are several end-to-end designs for R, t computations [214], [321]. Although the predicted depth map can be utilized to recreate the 3D world via fusion of points, as is done in [162], most current works just do depth map estimation [306], [361].

In the literature, there are two widely used techniques for training pose and structure estimation: the self-supervised learning paradigm and the supervised learning paradigm.

1) Supervised learning: By reducing errors in estimating the camera position in comparison to the posture labels of the camera, one can trains a CNNs in a supervised fashion. Since CNNs were, formerly, utilized for classification purposes, pose estimation was initially defined as the problem of classifying the values on the quantized space of the camera's postures in the 3D space. It's likely that Konda and Memisevic [361] were the first to suggest utilizing this concept to estimate visual odometry. They used a stereo camera to forecast the R, t.

Konda et al. [151] utilize synchronous autoencoders to learn the motion and depth from stereo pairs. In order to predict the orientations and the velocities in a classification paradigm using softmax, the depths and motions features are processed by a CNN. DeTone et al.[61] suggested "HomographyNet" which learns two-frame homography via parameterization of the homography in a 4-point fashion rather than estimating general motion comparable to basic matrix. A classification paradigm, which is trained using cross-entropy loss, and a regression paradigm, using a Euclidean loss, were the two networks they proposed. Since the estimation here is naturally continuous, they demonstrated that the regression paradigm is better than the classification paradigm in accuracy.

All contemporary methods for R, t estimation utilize regression-based CNN because it was shown that regression is capable of accurately solving the problem. Mohanty et al.[216] used a pretrained AlexNet network [154]. To regress the R, t using a fully connected layer, the two frames are input to two concurrent AlexNets, and the outputs are then concatenated. They concluded from their research that the derived AlexNet features are not universal for the issue of VO. As a result, the odometry delivers acceptable results only if the distribution of the test and train are similar.

Odometry estimation cannot be done using the pretrained encoders used to detect objects in the scene in a classification paradigm, thus researchers utilize the networks which are designed to estimate optical flow to generalize the learnt parameters in various contexts. "Flowdometry," a network created by Muller and Savakis [220] is one of them. It consists of two successive CNNs, the first of which predicts optical flow and the second of which calculates camera motion. Both networks use the FlowNetS [65] architecture, however the second network substitutes a fully connected layer to accommodate inter-frame odometry calculation.

An end-to-end CNN was created by Melekhov et al.[214] to calculate ego-motion between two viewpoints. To process the input frames at the same time to preserve the spatial

information in the feature tensors, they utilize two CNNs in parallel which share their weights before adding a spatial pyramid pooling layer. At the end they add two FC layers to forecasts camera translation and rotation.

RCNN, a hybrid of CNN and RNN, is how Wang et al. [321] develop "DeepVO". It is a network which is able to learn sequential motion models in a dynamic scene from a video and is trained in an end-to-end fashion. Formerly, the scene was simply represented geometrically by CNNs. On the other hand, RNNs are designed to learn temporal information, like speech or language [166]. They are capable of doing so since they keep the record of every element of the temporal information. It comes to light that combining CNN and RNN yields noticeably improved results in VO and deliverers competitive results when compared to state-of-the-art approaches (See [87]). A deep learning system would not be able to deal with repetitive objects in front of camera, which may reduce posture estimate accuracy.

Another challenge in estimating depth map are semi-transparent surfaces and surfaces which reflect the light. While the light field methods are able to deal with these cases very well, the depth estimation methods show very poor performance in the scenes which contain surfaces with these traits. The reason behind this is the fact that most depth estimation algorithms assume one true value for a pixel although there might be several different semi transparent surfaces which contribute to that pixel. Exploring this issue, the main idea in [172] is estimating a posterior depth distribution instead of a single depth value. Based on this idea several algorithms based on deep-learning explored and developed in [172] to address the issue.

MDE can be defined as a omni-directional problem as well. [184] addresses the omni-directional MDE. More specifically, they try to solve the challenge of utilizing deep-learning algorithms in omni-directional MDE problem which accepts RGB images which contain dramatic spherical distortion. To this aim, their algorithm extract perspective 2D

pathches out of the omni-directional RGB image which has less spherical distortions. Then they cast the problem as a multi-view depth estimation problem using CNN. However, to reduce the inconsistency between these patches, they introduce a geometry-aware feature fusion mechanism. This mechanism fuses 2D features from images and the 3D geometric features to decrease the inconsistency between patches. Next, they utilize the self-attention transformer module to aggregate the information from different patches. This step also helps the previous fusion step to makes the consistency between the patches better. At the end, they makes the estimated depth better in an iterative fashion using the more accurate geometric features.

2) Unsupervised learning: A CNN in MDE may be trained in a self-supervised fashion in case where the ground truth is not available or it is expensive to attain. This is done by reducing the photometric error in a manner akin to LSD-SLAM[69]. The technique was created by Zhou et al.[361] using this principle that one can synthesize a target frame using a source frame and the R, t between the two frames. Then comparing between the synthesized target frame and the real target frame one is able to train a network in MDE. Vijayanarasimhan et al. [306] created a three-dimensional scene-flow, instead, utilizing the camera motion prediction, the dynamic object segmentation, and the depth map estimation. The algorithm uses a convolutional-deconvolutional network.

Luo et al. [202] enforce geometric constraints on the video's pixel values using a traditional SfM reconstruction. They use a learning-based prior, or a CNN trained for single-image depth estimation, as opposed to the ad-hoc priors used in classical reconstruction. When put to the test, they fine-tune this network to satisfy the geometric constraints of a specific input video while preserving its capacity to generate depth features in less constricted areas of the video. They claim that even hand-held recorded movies with a significant amount of dynamic motion may be handled by the system.

Another unsupervised depth estimation is the algorithm suggested in [126]. The network is called RM-Depth and designed to jointly learn an un-constraint object motion, ego-motion

and depth map. Depth map is estimated using Recurrent Modulation Unit which fuse the encoder and the decoder in an iterative and adaptive fashion. They also utilize residual up-sampling to learn edge-aware filters. Most importantly, they recover a 3D motion field consist of moving objects in the scene. They do not use any segmentation labels. However, they still use one rotation and translation for the whole scene.

Self-supervised depth estimation can be interleaved in a self-distillation algorithm to get supervisory signals. To this aim, [236] designs a self-distillation and self-supervised monocular depth estimation network to learn depth estimation. First, the author trains their network in a self-supervised fashion on super-resolution RGB images based on reconstruction loss. Then they use scale invariant logarithmic loss and the pseudo labels from the trained network in the last step to retrain the network. To solve the problem of scale consistency between different frames the author utilizes a technique [337] to compute the scale factor. The scale consistency module works by estimating the ratio between the real camera height and its estimated value.

[97] exploits multi-frame paradigm instead of single-frame paradigm to improve depth estimation. In this way they are able to benefit from geometric connection between consecutive RGB frames in a video through feature matching on top of the learning appearance based approach. They utilize feature matching in a self-supervised manner to estimates monocular depth. They suggest a transformer-based structure to generate their cost volume. Specifically, they design a depth-discretized epipolar sampling module to select among the matching candidates. Then they refine the depth predictions through self-attention and cross-attention modules. In this way, they make the matching probability more efficient than the standard similarity metrics which tend to get stuck in local minima. Finally, the result of the above-mentioned operations deliver depth estimation using a decoder. The model is trained in an end-to-end fashion using photo-metric loss. Application of multi-view depth estimation and transformers is not limited to only depth estimation. For example [32] estimates depth in a multi-view fashion for novel view synthesis problem.

[17] addresses two issues in multi-view geometry. First, the high memory consumption of multi-view cost-volume, which in turn result in slow inference as well, second, difficulties in multi-view matching due to moving objects in the scene, reflective surfaces and texture-less ones. So they propose to fuse single-view MDE into multi-view geometry, to benefit from the efficiency, robustness and accuracy of multi-view MDE. To this aim, they estimates a single-view depth a pixel-wise Gaussian probability distribution for each frame. Then they sample the distributions to create adaptive candidate samples. This adaptive method results in more efficiency and accuracy. They also use a matching score paradigm to make sure that the predicted mutli-view and single-view depths are consistent.

Depth-from-focus is another technique which can be used to estimate depth. When focus of a camera changes, it creates a stack of images with focus at different depth. This can create a supervision signal to train a network provided that the stack of images are available. In [341], a CNN is suggested to estimates the best focused values for each pixel in the focal stack under consideration. Then the depth can be estimated using the estimations in previous step.

4.3 Motion segmentation and tracking of dynamic objects in 3D

Motion segmentation in dynamic scene and 3D tracking classify objects according to their motion and follow their 3D trajectories. The flowchart of these methods is shown in Fig. 4.5. As depicted in the Fig. 4.5, all the features extracted from the frame as well as optionally dynamic features are utilized to segment the scene into different moving objects in approaches which are feature-based. The methods based on deep-learning, however, has the capability to deal with the visual frames automatically. The 3D tracking module is then fed with the segmented dynamic objects to produce the object trajectories. It is optional to use camera rotation and translation as well as the three-dimensional point-cloud acquired from A.2 in Fig. 4.5 to aid in the tracking procedure. The trajectories of objects

are coherent with the background (i.e. static) environment because of the utilization of the three-dimensional point-cloud. These algorithms are covered in this section. See Fig. 4.5.

4.3.1 Segmentation of scene based on dynamic objects

Segmentation of a scene into moving objects, referred to as monocular motion segmentation, multibody motion segmentation, or eorumotion segmentation [143], [257], [286], [258], groups matched features in the scene into areas in the scene which belongs to the same moving objects. Because of the problem's chicken-and-egg nature, it is very challenging. The features must first be clustered into motion models. However, all moving objects must have motion models for the features to be clustered. Presence of outliers, noise or missing matched features as a result of occlusion, noise, motion blur, or tracked features which are lost all contribute to the issue. Dealing with degeneracy in motion models, which occurs if one object travels in the camera motion plane, or in the camera motion direction, and at the same speed as the camera, is another issue. Dependent movement (for example, when a pair of objects travels together in 3D, or articulated motion) is another one as well. The methods which are in use to solve segmentation of the scene based on dynamic objects are covered in this section.

Statistical model selection

One motion model can be used to describe how a static scene's features change from one image to the next. In contrast, the feature which are dynamic come from multiple motion models, each of which is connected to a distinct moving body. Essential matrix, Fundamental matrix, projectivity/homography, affine fundamental matrix or affinity are the possible mathematical way of expressing the motion models. The goal is to fit all the features in the scene into them most effectively.

Two simple statistical techniques to fit the data to the above-mentioned models are using RANSAC [72] and the Monte-Carlo sampling iteration [267]. These methods create a set

which will be used as inlier and dismiss the remaining data to fit a model to them. The outliers are then sampled once more in order to fit a new model that represents majority of the samples. This process is repeated until a threshold is met, i.e. the error is low enough. Also, one can start over with this motion segmentation technique to obtain a large number of candidate models. Refer to Fig. 4.6 (b).

An information criterion is used to choose the scenario that is the best representative of the samples. The literature contains a number of these information criteria. One of them is maximizing the likelihood function while minimizing the flexibility of the model is chosen based of Akaike's information criterion (AIC) [4]. According to a certain metric, like Sampson distance approximation or reprojection error [112], the likelihood function is typically approximated to optimize the possibility of the observed correspondences. The model with the lowest AIC is then chosen by AIC.

Despite being widely used, this method lacks consistent estimates asymptotically and is susceptible to over-fitting. The reason is it does not consider the amount of observations. Schwarz [272] suggests the Bayes Information Criterion, a refinement algorithm based on the Bayesian theorem (BIC). By simulating the prior based on its complexity, BIC increases the posterior probability of viewing the samples. In contrast, Rissanen [253] created minimum description length (MDL) by reducing the data's coding length by utilizing a minimum-bit representation. By considering the quantity of observations and the size of the model, Kanatani [142], [141] introduced the Geometric Information Criterion (G-AIC, also known as GIC) in response to the limitations of earlier efforts. Geometrically robust information criterion (GRIC), developed by Torr [290], is another extension based on BIC, adding resilience to outliers and the ability to handle multi dimensions.

Schindler et al. [269] expanded the methodology to include multiple frames using essential matrix E, whereas earlier methods, like [266], only function as a two-frame technique. Schindler et al. [268] develop generalization of this method to other motion models and

camera models. Ozden et al. [231] consider practical factors. They dealt with how to break a cluster into two or how to blend one motion model with the static scene.

In another effort, the problem of model selection is presented by Thakoor et al. [286] as a combinatorial optimization. AIC is used as the cost function, in this technique. Using the branch and bound approach the problem is broken into smaller sub-problems. The segmentation are produced via local sampling of correspondences, and to account for outliers the null hypothesis is included.

Sabzevari and Scaramuzza [257] used the projective trajectory matrix framework's factorization to apply a statistical model selection technique. Reprojection error is utilized to weed out unreliable proposals while motion models are created using epipolar geometry. By repeatedly doing so, they improve their predictions. Differently, [258] expand the problem in such a way that the problem's computations can be done using the two-point approach [229] and the one-point algorithm [264], [265].

Sub-space clustering methods

The Sub-space clustering is a general technique for clustering low-dimensional sub-spaces that Kanatani [141] introduced which its application is not limited to motion segmentation. Its development is founded based on the insight that some set of low-dimensional sub-spaces can represent high-dimensional data samples. The segmentation of the dynamic scene under the subspace clustering framework is essentially locating each of these sub-spaces and associate them with moving objects (see Fig. 4.6 (a)). Nevertheless, as these sub-spaces and segments are not known, it is necessary to estimate the sub-spaces and cluster the samples to distinct sub-spaces simultaneously. Gear [84] and Costeira-Kanade [52] address this issue utilizing the discovery that the space of the rigid moving objects is a linear sub-space and it is possible to recover each linear sub-space by enforcing the rank

requirement. A wide variety of approaches are done in this category [305, 339, 93, 303, 68, 250, 340, 193, 39] among which [304, 84, 355] are online clustering techniques.

Dynamic object segmentation using deep-learning

The motion segmentation problem may be solved with assuming predetermined number of rigid motion models utilizing DNNs. Producing dense object masks and its related cost functions may be done using optical flow or three dimenstional point-clouds. Byravan and Fox propose "SE3-Net" that can segment preset number of models expressed in SE(3) transformations from a three-dimensional point-clouds in their paper [29]. The network is a convolutional-deconvolutional encoder-decoder network. Two parallel networks, one of which is a CNN that produces the masks for motion models and the other one is built using FCs that produces SEs. For more details refer to [29].

According to Vijayanarasimhan et al. [306], optical flow can be used to segment dynamic objects using DNN. They created the "SfM-Net". It is a geometry-aware network with the ability to predict ego-motion, structure and motion segmentation of the scene. Two stream convolutional-deconvolutional sub-networks that serve as the structure and motion networks make up the model. The motion model calculates static and dynamic motion models, the structure network learns to anticipate depth. The point-clouds from depth predictions is then warped based on the motion models and then reprojected back into the photo space to create optical flow. While fully supervised-learning is also possible, this method allows the network to be trained in self-supervised manner by minimizing photometric error.

Casser et al. [34] propose an approach which is able to model moving objects. The main idea is to introduce geometric structure in the learning process, by modeling the scene and the individual objects; camera ego-motion and object motions are learned from monocular videos as input. Furthermore an online refinement method is introduced to adapt learning

on the fly to unknown domains.

Appearance-based detectors achieve remarkable performance on common scenes, benefiting from high-capacity models and massive annotated data, but tend to fail for scenarios that lack training data. Geometric motion segmentation algorithms, however, generalize to novel scenes, but have yet to achieve comparable performance to appearance-based ones, due to noisy motion estimations and degenerate motion configurations. To combine the best of both worlds, Yang et al. [343] propose a modular network, whose architecture is motivated by a geometric analysis of what independent object motions can be recovered from an egomotion field. It takes two consecutive frames as input and predicts segmentation masks for the background and multiple rigidly moving objects, which are then parameterized by 3D rigid transformations.

Optimization for motion segmentation

Ranftl et al. [247] offer an algorithm for dense depth estimation from a single monocular camera that is moving through a dynamic scene. The approach produces a dense depth map from two consecutive frames. Moving objects are reconstructed along with the surrounding environment. They provide a motion segmentation algorithm that segments the optical flow field into a set of motion models, each with its own epipolar geometry. Then they show that the scene can be reconstructed based on these motion models by optimizing a convex program. The optimization jointly reasons about the scales of different objects based on an ordering constraint in the scene as well as a smoothness constraint and assembles the scene in a common coordinate frame, determined up to a global scale.

4.3.2 Dynamic objects' 3D tracking

It is difficult to track the moving objects in three dimensions even if one knows their position and the depth of each point in the three dimension scene. The difficulty emanate from the triangulation method, which is the commonly practiced method to determine

the mapping ([113]), can not be utilized to track the objects since the beams which are projected back to the three-dimensional world from the matched points in the frames do not intersect. The three-dimensional points X should be calculated by finding these intersections of the beams of x_1 and x_2 via their corresponding camera poses, P_1 and P_2, given x_1 and x_2, the matched points in frames 1 and 2. See Fig. 4.3. To address this issue, alternative methods are needed. The methods that are currently used to recover the three-dimensional world trajectories of the dynamic objects are covered in this section.

Trajectory Triangulation

Since the above-mentioned beams do not intersect in the case of moving objects, basic triangulation [113] is not effective to map the dynamic scene. However, when it is known that one object is physically constrained to fulfill a specific mathematical form, Avidan and Shashua [14], [15] invented the technique "trajectory triangulation" which reconstruct the three-dimensional world points that belong to a dynamic object and the scene. They assume that the point is traveling along an unidentified line in the three-dimensional world. Then finding the parametric line that meets the correspondence from several viewpoints is thus added to the reconstruction task. At least 5 frames are necessary to find the solution uniquely. Shashua et al. [273] considered that the item is travelling along a conic section as opposed to along a straight line.

While Kaminski and Teicher [139], [140], formulate the "trajectory triangulation" using a family of polynomial curves to convert the non-linear trajectories problem to a linear one, Park et al. [234] modeled it as a linear combination of trajectory basis vectors to manage missing data, making it possible to predict the recovery of 3D points with confidence using least squares. They established a criteria, called "reconstructability", that enabled a precise reconstruction of the three-dimensional scene [235].

Particle filter

Tracking of dynamic objects in three-dimensional world in monocular setting is known as the Bearing-only-Tracking (BOT) problem since it cannot be detected that there is a distance from the target to the observer (observability issue). For this problem, a solution which is based on filtering is desirable because it can model the uncertainty of the observer's and the target's position and velocity [3, 165]. Particle filters were used by Kundu et al. to predict the speed and position of the moving objects [158].

4.4 Simultaneous reconstruction and motion segmentation

Factorization allows for simultaneous multibody motion segmentation and reconstruction of the 3D structure of dynamic objects. The motion models of the segmented features as well as their 3D structures are produced via dynamic object segmentation and reconstruction given the feature correspondences. The procedure for this joint motion segmentation and reconstruction task is shown in Fig. 4.7. In general, the output from applications A and B may be merged to achieve a comparable outcome to this approach, even though factorization can create both segmented objects and their 3D structures.

4.4.1 Factorization

Undoubtedly one of the most well-known SfM strategies is factorization. It can concurrently handle the segmentation and reconstruction problems and has a beautiful mathematical formulation. Based on the rank theory, it was initially developed by Tomasi and Kanade in [288]. In general, there are two different branch of MDE in dynamic scene stem from Factorization.

Multibody structure from motion (MBSfM)

Multibody Structure from Motion (MBSfM) extends conventional Structure from Motion (SfM) for a rigid camera motion into n rigid bodies of motion. To tackle the MBSfM problem under the affine camera model, Costeira and Kanade [52] developed the "shape interaction matrix", a mathematical construct of object shapes that is independent of object motion and the coordinate system which has been selected. It was discovered that this "shape interaction matrix" preserves the original subspace structure. Assume $\bar{W} = U\Sigma V^T$ is the rank-r SVD decomposition of measurement matrix such that $U \in \mathbb{R}^{2f\times r}$, $\Sigma \in \mathbb{R}^{r\times r}$, and $V \in \mathbb{R}^{p\times r}$. Then the "shape interaction matrix" Q is defined as

$$Q = VV^T \in \mathbb{R}^{p\times p} \tag{4.1}$$

Equation (4.1) has the intriguing characteristic that the entry is 0 if feature trajectories a and b belong to separate objects. Kanatani has mathematically shown this characteristic [141]. On the basis of this discovery, motion segmentation and reconstruction may be accomplished by sorting and thresholding the entries of Q.

Costeira and Kanade [52] cluster the whole structure by maximizing the sum-of-squares entries of a block diagonal matrix under the restriction that each block indicates a moving objects. Ichimura [128] used a discriminant criteria [230] to divide the sorted rows of Q into miscellaneous motion models that maximize separation between sub-spaces. Gear [84] demonstrated that instead of clustering the subspace using SVD, echelon canonical form gives direct information on the grouping of points to the sub-spaces.

The projective depths are recovered by Sturm and Trigss [282] via the calculation of epipoles and fundamental matrices. The factorization based on a perspective camera was expanded by Hartley and Schaffalitzky [111] to include missing and ambiguous data. To approximate missing data with a low-rank matrix, they created an iterative power factorization approach.

Li et al. ([183]) alternate among motion segments using subspace separation and projective depth estimation to achieve convergence. Minimizing the reprojection errors is followed by iterative refining to estimate the projective depth. Instead, Murakami et al. [223] attempted to eliminate the need for depth estimation by proposing depth-estimation-free circumstances. If two conditions are satisfied, initial values does not need to be computed.

Nonrigid structure from motion (NRSfM)

Bregler et al. were first ones to employ a scaled orthography camera model in citebregler2000recovering to present the Nonrigid Structure from Motion (NRSfM) approach, which is based on Tomasi-Kanade factorization. A nonrigid object was represented as a k key frame basis set $\{Bi\}^k i = 1$ with each Bi denoting a $3 \times p$ matrix representing p feature points. This basis set's linear combination creates a particular configuration to the extend that $B = \sum^k i = 1 li.Bi$, where $B, Bi \in \mathbb{R}^{3\times p}$ and $li \in \mathbb{R}$. They normalize the feature points just like [288] and removing the translation vector. As a result the measurement matrix becomes

$$\tilde{W} = NB = \begin{bmatrix} l_{11}R'_1 & \cdots & l_{1k}R'_1 \\ \vdots & \ddots & \vdots \\ l_{f1}R'_f & & l_{fk}R'_f \end{bmatrix} = \begin{bmatrix} B_1 \\ \vdots \\ B_k \end{bmatrix}, \tag{4.2}$$

Here, R' stands for the first two rows of the rotation matrix R (the last row of R can be inferred by computing the cross-product of the first and second rows of R due to the orthogonal projection of the orthographic camera model). The SVD can be employed to factorize $\tilde{W}$ by selecting the first $3k$ singular vectors and singular values. By rearranging N's elements and factorizing it with SVD, the estimated rotation matrix $R'f$ and the shape basis weights li can be extracted from the data set. In order to create a matrix G that maps $R'f$ and Bk into an unique solution [26], orthonomality constraints are lastly applied.

Basis constraints are introduced by Xiao et al. [333] as an alternative to orthonomality constraints so that the nonrigid factorization problem can be resolved in a closed-form solution. The motion matrix is projected onto the manifold of matrix constraints by Paladini et al. [232] instead of directly enforcing the metric constraints; as a result, the factorization can be carried out iteratively using least squares. On the other hand, Akhter et al. [5] suggest a dual approach by outlining a method based on trajectory space that eliminates the requirement to compute basis vectors. The body motions are compactly described using the Discrete Cosine Transform (DCT).

A recent proposal by Kumar et al. [155] to combine MBSfM and NRSfM into a multibody nonrigid deformations system was made. The feature trajectories were treated as the union of various linear or affine subspaces. It makes it possible to use the alternating direction approach of multipliers to jointly optimize nonrigid reconstruction and nonrigid motion segmentation (ADMM).

For more details about dynamic scene MDE one can refer to [260].

4.4.2 Deep-learning based nonrigid structure from motion

Li et al. [177] present a method for jointly training the estimation of depth, egomotion, and a dense 3D translation field of objects relative to the scene, with monocular photometric consistency being the sole source of supervision. They show that this apparently heavily underdetermined problem can be regularized by imposing the following prior knowledge about 3D translation fields: they are sparse, since most of the scene is static, and they tend to be piecewise constant for rigid moving objects.

Closely related to MDE in dynamic scene with nonrigid motions in the scene using deep learning is [244]. In this paper the Authors introduce D-NeRF, a method that extends neural radiance fields to a dynamic domain, allowing to reconstruct and render novel images of objects under rigid and non-rigid motions from a single camera moving around the scene.

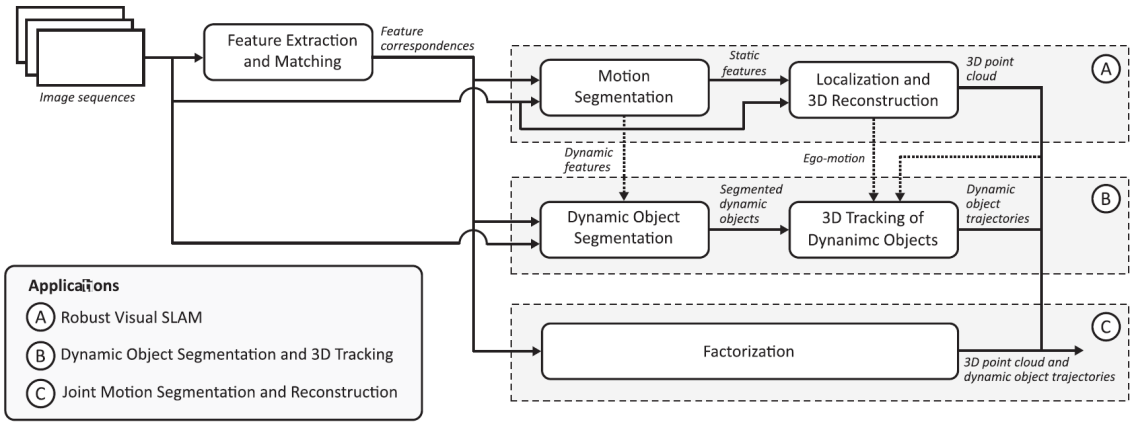


Figure 4.1: Block diagram which depicts the general pipeline in visual localization and 3D reconstruction in dynamic scene. Figure from [260].

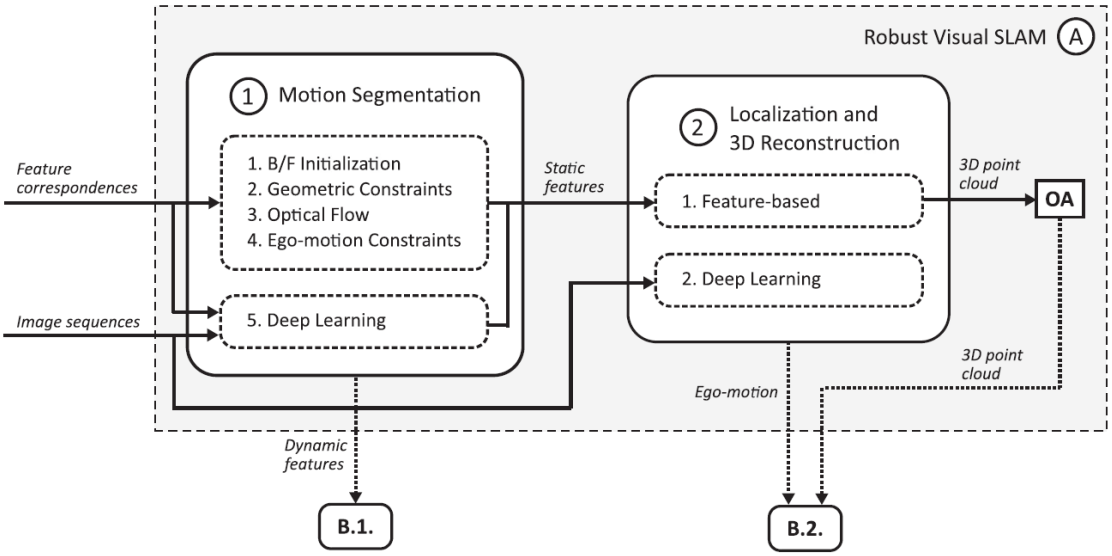


Figure 4.2: Block diagram of general algorithms in the section robust visual SLAM/SfM. Figure from [260].

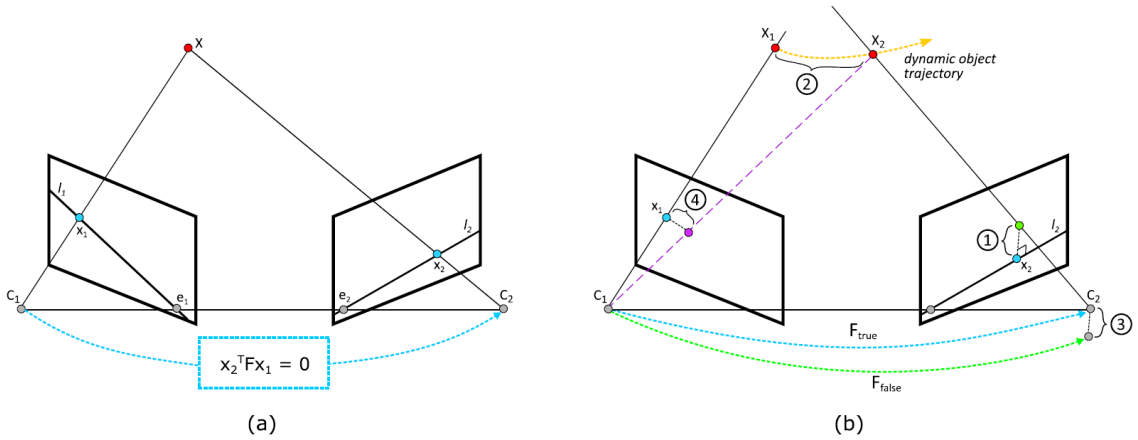


Figure 4.3: (a) In a static scenes, epipolar constraint $x_2^T F x_1 = 0$ constrains the image point from x_1 to x_2. (b) In a a dynamic scene the epipolar constraint is violated. Figure from [260].

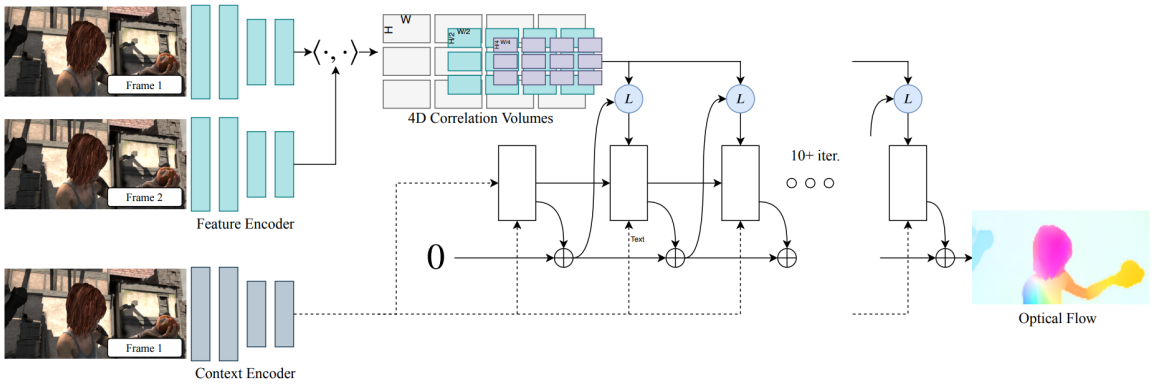


Figure 4.4: Overall structure of the RAFT [285]. Figure from [285].

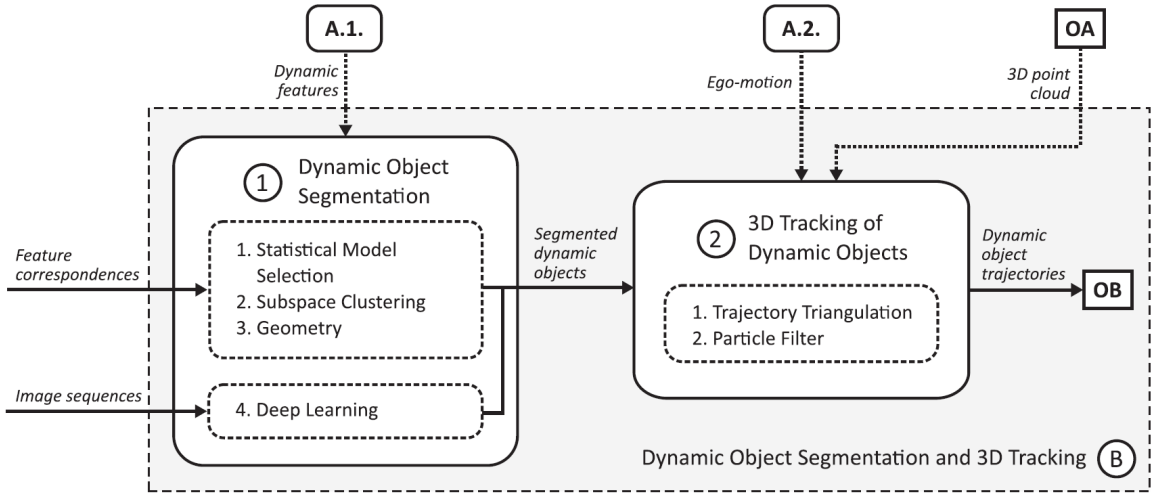


Figure 4.5: Block diagram of a general motion segmentation, and 3D tracking. Figure from [260].

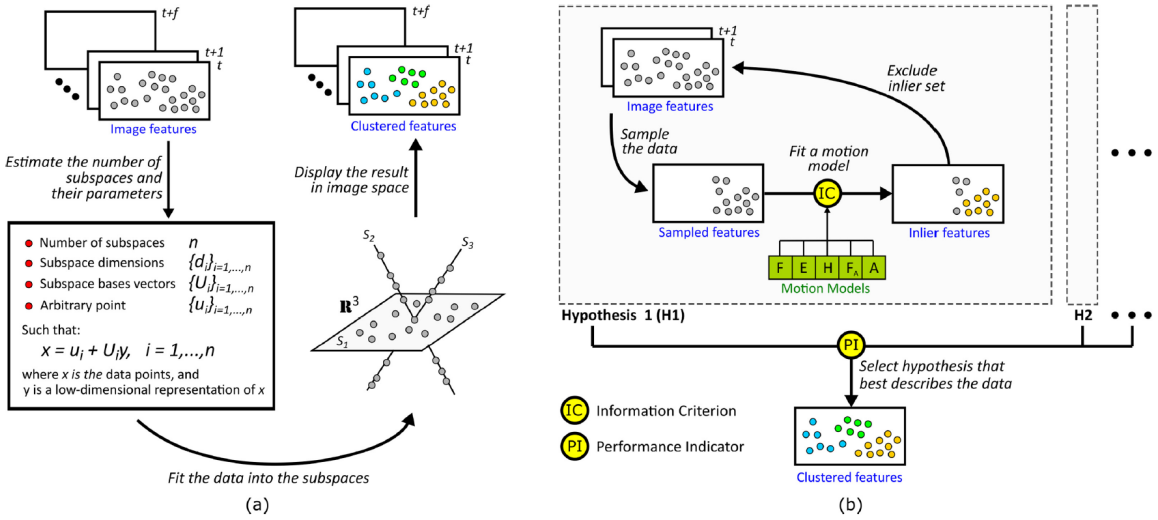


Figure 4.6: An example of dynamic object segmentation using (a) sub-space clustering and (b) statistical model selection. Figure from [260].

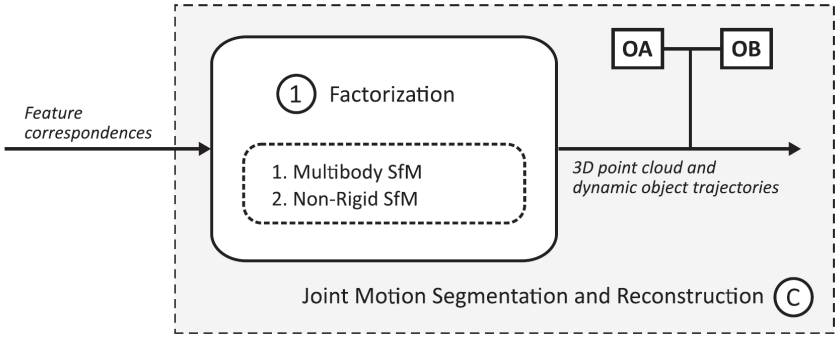


Figure 4.7: Block diagram of a general joint motion segmentation and reconstruction. Figure from [260].

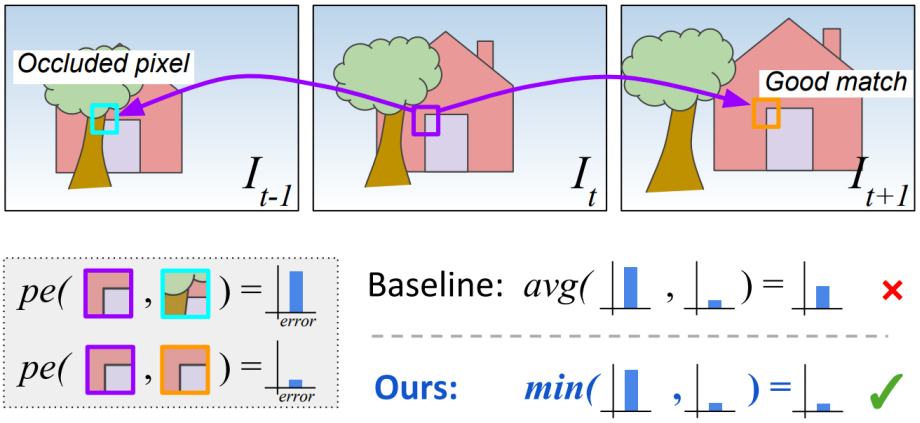


Figure 4.8: Minimum error suggested by [92] instead of average error can potentially detect occlusion. However it needs three views of the scene.

Chapter 5

Explicitly Modeling Flexible Objects and Dynamic Scene in MDE

Broadly speaking, there are two effective (passive) approaches available to estimate the depth map of a scene. Traditional computer vision methods that rely on assumptions on camera models which result in pure geometric approaches, and deep learning based algorithms which consider a universal function, usually based on Deep Neural Networks (DNNs), and train it on an already recorded datasets, i.e. train dataset. The later methods rely on similarities of the distributions of the data in the already recorded train datasets and the unknown test datasets.

Deep learning based methods have remarkably improved the performance in many computer vision tasks including depth estimation. Although fine tuning the estimated depth maps using a sequence of frames, like bundle adjustment [296], is a well-known approach to improve accuracy, the methods are usually rely on monocular cues and done in single image fashion in inference time. In addition, single image depth estimation results can be used to initialize approaches that fine tune the depth using temporal information in the train of video frames.

5.1 Rational and challenges

Despite easiness of single image MDE vs other methods in depth map estimation, like stereo vision, there are inherent challenges exist in MDE

- Single image MDE is an Ill-posed problem
- Single image MDE is scale-ambiguous
- Estimation is difficult in low-featured area of the image

Single image MDE based on deep learning in general and especially the algorithm disscussed in chapter 3 address these issues. First, the single image MDE Ill-posedness is addressed in deep learning based MDE since the prior knowledge helps the network to retrieve the lost information, i.e. the 3D dimension or the depth map, which is lost at the time of capturing the image. This prior knowledge helps with scale ambiguity of the paradigm in supervised case. In fact, the model learns the scale. In the method I designed in chapter 3, I discussed a novel approach to improve accuracy of single image MDE based on geometric attention. The algorithm especially addresses the issue with feature-less area of the image. It does that since it uses attention to help the network learns better features in encoder using the depth features. See Fig. 1.7 and 3.2.

However, gathering ground truth depth and synchronizing it is an expensive procedure. So the algorithm needs prior knowledge or another source of supervisory signal. These have been the incentive for researcher to think about unsupervised/self-supervised MDE algorithm. The temporal information which exist in a train of images captured in a video can be exploited as the source of supervision using epipolar geometry (5.3) and PnP (5.6). Designing an algorithm that exploit the temporal information instead of ground-truth depth as the source of supervisory signal in MDE in dynamic scene is what is addressed in this chapter.

In addition to the above-mentioned challenges in the MDE problem, there are specifically other issues need to be addressed in an unsupervised MDE paradigm

- Dynamic scene
- Flexible objects

Consider a scenario in which during the time that a frame is captured until the next frame is captured the objects in the scene move independently from each other. This is called a dynamic scene and computer vision methods which model only the static scene are not enough to accurately model a dynamic scene. The other challenge is modelling flexible objects. As it is discussed in section 5.3 and section 5.4 the geometric pinhole camera model is designed to represent a single object which is rigid. At the same time, many moving objects like animals, humans, and even a bag blown into the air are flexible objects. As a result, it is necessary to explicitly utilize a flexible model in a dynamic scene to accurately model the scene. See Fig. 5.1 and Fig. 5.2.

Despite the recent advances in visual SLAM, Structure from Motion, and unsupervised MDE algorithms in dynamic environments, each proposed approach comes with advantages and disadvantages. Many of the aforementioned methods are well-developed methods by computer vision and robotics communities. They are sharing two traits. First, all of them require direct or indirect feature correspondence as input. See Figures 4.1, 4.2, 4.5, 4.7. Secondly most of them, specifically multi-view multi-body geometry constraints were developed before the deep learning algorithms show their effectiveness in extracting abstract non-local, automatic features/information from RGB images.

Appearance-based (direct, deep-learning based) methods achieve remarkable performance on common scenes, benefiting from high-capacity models. Although, deep learning based method are very successful in understanding the whole scene in a dense fashion and being fairly light considering the algebraic nature of this type of depth estimation, they suffer from 1) accuracy 2) poor generalization when the test and train distributions are not close.

They are not able to accurately predict the scene with all of its details when the test dataset distribution is far from the train dataset. In fact, they might even fail. In addition, they rely on monocular cues which can be exploited for adversarial attack in security or safety systems [338].

On the other hand, traditional geometric methods based on optimization, tend not to be sensitive to the above-mentioned issue since they essentially do not rely on any prior knowledge, i.e. distribution of any training data. Instead, they benefit from temporal information of sequences of images/videos or synchronous camera rigs to extract the depth in an unsupervised manner.

At the same time, explicitly modeling of a dynamic scenes as well as flexible objects in monocular depth estimation using traditional computer vision methods is a big challenge. The reason is lying on the inherent fashion of estimation: scene might changes in a flexible and dynamic way between two consecutive frames. It should be noted that deep learning based methods can handle them to some extend since they usually estimate the depth using single image, not always though. The down side for single image depth estimation is loss of accuracy though.

In addition, single image depth estimation is an ill-posed problem. That is, it is not mathematically possible to uniquely estimate the 3rd dimension (or depth) from a single 2D image. On the other words, going from 3D world to 2D images is a one-way function which is irreversible directly. During this process the information of the scene related to the 3rd dimension is lost. Deep learning based method solve this using pre-trained model on pre-recorded train datasets.

Considering weakness and strength of each of these two approaches, a hybrid methods which benefits from both good generalization of geometric methods by extending traditional geometric models ability to handle flexible and dynamic objects in the scene and

interleaved it with deep learning networks to create a self-supervised training pipeline is the aim of this chapter.

It is not as easy as one might think though. The reason behind that is when you are working with sparse correspondence for an image of typical size 480×640 even if they use the best feature based extraction methods like AKAZE [6], they get around 1000 reliable points while the dense correspondence have $480 \times 640 = 307200$ almost 300 times more data to be processed. As a result adapting the sophisticated traditional computer vision algorithm to the hybrid system so that it would be able to process the data in real-time is a necessity [247].

On top of that, one might question the benefit of designing monocular depth estimation algorithms while stereo vision can avoid many challenges indigenous to MDE like scale ambiguity or problems like dynamic scene as well as difficulties in exploiting monocular cues. To answer this question, it should be noted that the stereo vision is limited to approximately 10 meters similar to D435 (an RGBD camera). This limitation springs from the sensitivity of the problem to the distance between stereo rigs. Farther than this distance, human vision mainly relies on monocular cues. So the first benefits is that the MDE has longer range than the stereo vision. On the other hand, having an efficient monocular algorithm can benefit many single camera devices available almost everywhere. The third benefit of monocular video depth estimation is that such algorithms pave the way for an efficient exploitation of temporal information in stereo video depth estimation. The motion model from one camera to the other camera in the stereo rig is a constant motion model while addressing each time stamp motion models of the scene to another timestamp ones require an algorithm similar to the topic of this chapter. Other benefits of using MDE could be less computational resources needed and avoiding the baseline issue as well as calibration difficulties of stereo rigs.

Contributions: To sum up, three innovations are suggested by this paper

Extending the geometric constrain to model full flexible scene: The first innovation of this paper is extending the geometric constrain to model the whole scene so that the model explicitly describe motion models at pixel level. As a result the model is able to model any flexible object without any assumption on the number of moving object in the scene.

Moving object detection loss: The second innovation is designing a motion model detection cost function which automatically detect moving object in the scene while considering flexibility of each object. In addition, this approach does not consider any assumption on the number of motion models or estimate it. Also, the method does not need to estimate the relative scale between the moving objects.

Synchrony theorem and Synchrony loss: The third innovation relates the different components of the pixel level motion models spatially to each other so that the model is able to explicitly model flexible objects at the same time be constrained enough. See Fig. 5.3 and Fig. 5.13. This is enforced by the Synchrony cost function. The cost function is supported by the Synchrony theorem. An outline of the proof is provided in this chapter.

5.2 Related works

There are 3 unsupervised approaches which designed to deal with dynamic scene.

5.2.1 Robust static methods

These methods ignore the dynamic nature of the scene. To compensate for this lack of exactness in modelling the scene, they use robust methods like RANSAC to decrease the effect of outliers and/or utilize deep learning networks to benefit from the robustness of the deep learning approaches [366, 153, 17, 99, 97, 177, 236, 126].

5.2.2 Explicitly modeling dynamic scene utilizing motion models estimation

This category of approaches first find the motion models in images. The motion models could be expressed as Essential matrices, E, or Fundamental matrices, F. This procedure is called motion segmentation. There are assumptions over number of moving object on the scene, though. In fact, the number must be estimated. Then depth for each motion model is estimated. Finding the relative value of the scales is a challenge in these approaches since scale is ambiguous [247, 34, 309, 298, 343, 290].

5.2.3 Factorization

These category of approaches estimates both the depth map and the motion models at the same time without any assumption on the number of motion models or any need to estimate this number. In fact they have solid mathematical background which provides the user with a closed form like multi-body multi-view structure from motion (MBSfM) [52]. Bregler et al. [26] extend the MBSfM to non-rigid motion estimation (NRSfM). However, the first work is considering affine model for camera instead of perspective camera model. In an affine camera model camera center is at infinity and it has zero perspective. The fundamental matrix in an affine camera model has 4 degree of freedom instead of 7 degree of freedom in pinhole camera model. Also, the NRSfM considers cartographic camera model. In case of a full pinhole camera model, optimization is required.

5.3 Relaxing static scene paradigm into dynamic scene and flexible object paradigm

Considering the weaknesses and the strengths of each of the approaches discussed in this chapter so far, a hybrid methods which benefits from both good generalization of geometric methods and robustness of deep learning is proposed here. The traditional

geometric models in computer vision handle mainly static scene. So it is suggested to relax it explicitly so that it is able to handle flexible and dynamic objects in the scene and interleaved it with deep learning networks to create a self-supervised training pipeline. See Fig. 5.4.

A preliminary result of relaxing static scene assumption in traditional computer vision methods to address flexible scene has been shown in Fig. 5.3. Two examples, (a) and (b), of the flexible model for two-frame structure from motion optimized on Sintel dataset. The input to this optimization are the two consecutive RGB frames depicted in Fig. 5.1 and Fig. 5.2, a target frame and the key frame. The target frame is the gray-scale depicted in top-left image in Fig. 5.3. Other than these two images the ground truth optical flow has been used in this specific optimization as well. However, in the algorithm (Fig. 5.4), the optical flow are extracted from the two RGB target and key frames using a CNN. See Fig. 5.4. In this algorithm, one motion model is considered for every pixel in the scene instead of only one for the entire scene. The algorithm utilizes two novel constraints, the "moving object detection loss" (See Fig. 5.12) and the "synchrony loss" (See Fig. 5.13.) to constrain the relaxed flexible scene model. In this way, it is able to change the optimization from an under constrained optimization to an over constrained optimization. k_x, k_y and k_z in Fig. 5.3 are the three component of n and t_x, t_y and t_z are the three component of t instead of only one n and t for the entire scene.

The overall training algorithm of MDE with explicit modeling of fully flexible objects and dynamic scene in an unsupervised manner is depicted in Fig. 5.4. The inputs to the algorithm are the two consecutive RGB target and key frames, I_t and I_k. The depth network, the n and t network in pink box, and the optical-flow network are all CNN networks to estimate n and t and the optical-flow from the I_t, I_k frames and the depth map from the I_k frame. The estimated depth is of size $H \times W$, the n and t are each of size $3 \times H \times W$ and the optical-flow outputs are the change of location of pixels in x, y directions in RGB I_t, I_k images (δx, δy) are of size $H \times W$ each. The cost function is in blue box

in Fig. 5.4. The main idea in the paper is to relax the motion models to pixel level. The "moving object detection loss" and the "Synchrony loss" are providing the optimization with appropriate constraints so that the optimization become an over constrained instead of being an under constrained. The innovations of this paper are depicted in color in Fig. 5.4. The algorithm can easily be extended to three frames or even longer sequences of RGB frames. The details are discussed in 5.4. Without further due, lets dive into the details.

5.4 Fully flexible dynamic scene algorithm

5.4.1 Preliminaries

Geometric static-scene and two-views computer vision methods are mainly based on pinhole camera model and the two criteria, perspective-n-point (PnP) and epipolar geometry. Assume

- p_t is a homogeneous point ($p_t = [u_t, v_t, 1]^T$) in frame I_t and p_k is the corresponding homogeneous point in frame I_k
- $R \in SO(3)$ is the rotation matrix in $3D$, Lie group, and $t \in \mathbb{R}^3$ is translation between the two frames I_t and I_k.
- D_k is the depth map of image k
- K is the intrinsic parameters of the camera which defines pinhole camera.
- $\theta = \|n\|$ and $\hat{n} = n/\|n\|$ where $\|\cdot\|$ represent norm 2.

The pinhole camera model is the linear transformation in $3D$ that takes each point in $3D$ world to the camera plane

$$P_{2D} = K\ P_{3D}. \tag{5.1}$$

Here

$$K = \begin{bmatrix} \alpha_x & s & x_0 \\ 0 & \alpha_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.2}$$

where α_x and α_y are the focal length of the camera in terms of pixel dimensions in x and y direction respectively and (x_0, y_0) is the principal point in pixel dimension. The parameter s is skew parameter and most of the times is zero.

The **Epipolar geometry** states that

$$p_t^T K^{-T} E K^{-1} p_k = 0,\ E = [t]_\times R. \tag{5.3}$$

Here, $[t]_\times$ is matrix representation of vector cross product

$$[t]_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix},\ t = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}, \tag{5.4}$$

and

$$[t]_\times v = t \times v\ \forall v \in \mathbb{R}^3, . \tag{5.5}$$

The **PnP** equation states

$$c\, p_t = KR\left[K^{-1} D_k p_k - t\right],\ c > 0. \tag{5.6}$$

So it relate p_k and its corresponding point upto a positive scaling factor c.

The relation between $R \in SO(3)$ and $n \in \mathbb{R}^3$ is simple

$$R = exp([n]_\times). \tag{5.7}$$

However, calculating matrix exponential is expensive. Instead it is a common practice to use Rodrigues' rotation formula. See Fig. 5.5 and Fig. 5.6.

One can project the triangle $AA'A''$ into plane P from Fig. 5.5. Since the rotation happens around the vector $\hat{n}$ then $\|OH\| = \|OH'\|$. So the vector OB which is the bisector of the angle HOH' is perpendicular to HH'. Then the angle $H'HH$" is equal to $\theta/2$. So

$$\|OH\| = \|\hat{n} \times v\| = \|\hat{n} \times (\hat{n} \times v)\| \tag{5.8}$$

$$\|HH''\| = 2\|OH\| \; sin(\theta/2) \; cos(\theta/2) = \|\hat{n} \times v\| \; sin(\theta) \tag{5.9}$$

$$\|H''H'\| = 2\|OH\| \; sin(\theta/2) \; sin(\theta/2) = \|\hat{n} \times (\hat{n} \times v)\| \; (1 - cos(\theta)) \tag{5.10}$$

$$v_{rot} = v + sin(\theta) \; \hat{n} \times v + (1 - cos(\theta)) \; \hat{n} \times (\hat{n} \times v) \tag{5.11}$$

$$v_{rot} = v \; cos(\theta) + (\hat{n} \times v) \; sin(\theta) + \hat{n} \; (\hat{n} \cdot v)(1 - cos(\theta)) \tag{5.12}$$

$$\hat{n} = \begin{bmatrix} \hat{n}_1 \\ \hat{n}_2 \\ \hat{n}_3 \end{bmatrix}, \; \hat{n}_1^2 + \hat{n}_2^2 + \hat{n}_3^2 = 1, \; \hat{n} \times v = \begin{bmatrix} 0 & -\hat{n}_3 & \hat{n}_2 \\ \hat{n}_3 & 0 & -\hat{n}_1 \\ -\hat{n}_2 & \hat{n}_1 & 0 \end{bmatrix} v = [\hat{n}]_\times \; v \tag{5.13}$$

So

$$R = I + sin(\theta) \; [\hat{n}]_\times + (1 - cos(\theta)) \; [\hat{n}]_\times^2 \tag{5.14}$$

and

$$R - R^T = 2sin(\theta)[\hat{n}]_\times, \; Tr(R) - 1 = 2cos(\theta) \tag{5.15}$$

Direct multiplication shows that

$$[\hat{n}]_\times^3 = -[\hat{n}]_\times. \tag{5.16}$$

So

$$exp([n]_\times) = exp(\theta[\hat{n}]_\times) = \sum_{k=0}^{\infty} \frac{(\theta[\hat{n}]_\times)^k}{k!} \tag{5.17}$$

$$= I + sin(\theta) \; [\hat{n}]_\times + (1 - cos(\theta)) \; [\hat{n}]_\times^2 = R \tag{5.18}$$

5.4.2 Cost function design

Cost function consists of

- Dense PnP loss
- Sparse PnP loss
- Dense epipolar loss
- Sparse epipolar loss
- Sparse optical flow loss
- Dense optical flow loss
- Moving object detection loss
- Synchrony loss
- Average size of t

among which the "moving object detection loss" and the "synchrony loss" are two of the innovations of this paper. The gradients are calculated based on weighted sum of the above-mentioned terms in the stochastic gradient descent optimization. The details are discussed in the experimental results section.

Dense PnP loss:

The dense PnP loss is designed to reduce the PnP error. This helps with the tuning of n and t network as well as the depth network and relate them to optical flow error. For this term assume

$$\Omega = \{(i,j)|i,j \in \mathbb{Z}, 0 \leq i \leq W, 0 \leq j \leq H\} \tag{5.19}$$

$$p_k = \begin{bmatrix} i \\ j \\ 1 \end{bmatrix}, \; p_t = \begin{bmatrix} i + \delta x(i,j) \\ j + \delta y(i,j) \\ 1 \end{bmatrix}, \; (i,j) \in \Omega \tag{5.20}$$

$$n = \begin{bmatrix} n_1(i,j) \\ n_2(i,j) \\ n_3(i,j) \end{bmatrix} \ , \ t = \begin{bmatrix} t_1(i,j) \\ t_2(i,j) \\ t_3(i,j) \end{bmatrix} \ , \ D(i,j) \ , \ (i,j) \in \Omega \tag{5.21}$$

Then one can use (5.6) and build the dense PnP loss.

$$\hat{p}_t = KR\left[K^{-1}D_k p_k - t\right] \tag{5.22}$$

However, the (5.6) relate p_t and p_k up to a positive scaling factor. So cosine similarity, $cossim(\cdot,\cdot)$, is utilized to build the error

$$\ell\left(error\right) = \ell\left(cossim(p_t, \hat{p}_t) - 1\right) \tag{5.23}$$

The L1 measure is used as ℓ.

Sparse PnP loss:

The sparse matched points are a good source to supervise training of the model in a self-supervised paradigm. "AKAZE" feature extractor is utilized to find matched points between the two frames I_t, I_k. See Fig. 5.8. However, there are two issues exist here. First, batching mechanism of $pytorch$ requires equal size of each single sample data used in one batch. Whilst the number of extracted matched points in different image pairs are random. It was solved by concatenating all the matched information from AKAZE feature matching algorithm and handling the error in python and $pytorch$. The second problem was more important. The matched points coming from AKAZE feature matching algorithm are not at integer grids (5.19).

$$p_k = \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} \ , \ p_t = \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} \tag{5.24}$$

while n, t and the D are still defined at the integer grid (5.19) and (5.21). So interpolating the values to floating points are required here. Assume that the values of a feature map f

at integer grid Ω (5.19) are available. Assume one is interested to find the value of f at arbitrary floating location (x, y). See Fig. 5.9.

The weighted sum of the four integer values around the floating point (x, y) is used as the interpolated value of the feature map f at the floating point (x, y). The inverse of the distances between the floating point and the four integer grid points around it are utilized as weights:

$$f(x, y) = \frac{\sum_{i=1}^{4} \frac{1}{d_i} f_i}{\sum_{i=1}^{4} \frac{1}{d_i}} \tag{5.25}$$

Note that the formula return the corresponding values at the integer grids correctly when the floating point (x, y) is at one of the integer grids.

Dense Epipolar loss:

The dense epipolar loss is designed to reduce the epipolar error. This helps with the tuning of n and t network and relate them to the optical flow error. Again, the integer grid Ω (5.19), $p_t,\ p_k$ (5.20) and $n,\ t,\ D$ (5.21) are the same as in the dense PnP loss. The error comes from epipolar geometry (5.3). Fig. 5.11 depict the actual implementation of the error.

$$\ell(error) = \ell(p_t^T K^{-T} E K^{-1} p_k),\ E = [t]_\times R. \tag{5.26}$$

The L1 measure is used as ℓ.

Sparse Epipolar loss:

The sparse epipolar term is identical in details to the dense epipolar loss but it is sparse and the p_k, p_t are at floating point locations instead of at integer grid Ω (5.19). They are just like (5.24). As a result, interpolating it to find the values of $f(x, y)$ from the values at integer grid Ω is needed. See Fig. 5.9.

Sparse Optical Flow loss:

Since the sparse matched points p_k and p_t from AKAZE feature matching algorithm are available, it is easy to benefit from that to create a supervision over optical flow network. The p_k and p_t are at floating points (5.24) and the interpolation over p_k and OF is needed. See Fig. 5.10. Also L1 measure is used as ℓ. See Fig. 5.10.

$$\ell(error) = \ell(p_t - p_k - OF) \ , \ OF = [\delta x, \delta y, 0]^T \tag{5.27}$$

Dense Optical Flow loss:

A dense source to supervise the optical flow network is required. Assume the sequence of the frames is a three parameters function f

$$f \ : \ (x, y, t) \longrightarrow \mathbb{R} \tag{5.28}$$

$$f(x, y, t) = I \tag{5.29}$$

where (x, y) is an arbitrary floating point on the image and t is timestamp of the image and I is the image intensities at the floating point (x, y).

$$(x, y) \in \mathbb{R}^2, \ 0 \leq x \leq W \ , \ 0 \leq y \leq H \ , \ t \in \mathbb{R} \tag{5.30}$$

Assume Δt is the time that takes the sequence of the frames to go from the frame I_k to the frame I_t. Assume the intensity of frames at two corresponding points during this time does not changes (which is a source of inaccuracy). Then one can expand f at the time k (frame k is represented by I_k) and around a point (i, j) at the integer grid Ω (5.19) using Taylor series expansion

$$f(i + \delta x, j + \delta y, k + \delta t) = f(i, j, k) + \frac{\partial f}{\partial x}\delta x + \frac{\partial f}{\partial y}\delta y + \frac{\partial f}{\partial t}\delta t + H.O.T \tag{5.31}$$

$$\left[\frac{\partial I}{\partial x}\right]_{i,j,k} \delta x(i,j) + \left[\frac{\partial I}{\partial y}\right]_{i,j,k} \delta y(i,j) + I_t(i,j) - I_k(i,j) \approx 0 \tag{5.32}$$

The optical flow network provides us with the δx and δy. The argument here is exactly the main optical flow heat equation in literature.

Moving object detection loss:

Assume one forces the gradients of each feature map components of the n, t go to zero. This means that all the feature maps in the entire spatial points, i.e. the integer grid Ω (5.19) represent only one number which, in fact, change the optimization into a static scene detection model. Now assume one lets some of the largest values in the gradients do not go to zero. Then the model can separates the parts of the images that want to move different than the static scene and different from each others. To this aim, the components of the n and the t are sorted in absolute value of their spatial gradients (quantiles) so that the extreme values do not forced to go to zero. These large exempt gradients are, in fact, the borders of moving objects. See Fig. 5.12. It also can remove noise. In this way, the n and t CNN is forced to detect moving objects in the scene. The L1 measure is used as the ℓ so the optimization is robust. As a result, it lets the object behave flexible based on the weight of the term.

$$q_\rho := quantile(\|\nabla f\|, \rho) \tag{5.33}$$

Here f is a feature map of size $H \times W$. Then

$$mask := \{(x,y) : \|\nabla f\| < q_\rho\} \longrightarrow \ell(mask \times \|\nabla f\|) \tag{5.34}$$

In practice, the partial derivatives in x and y direction are utilized separately in optimization in the components of the n and the t for stability reasons. See Fig. 5.12.

Synchronizing feature maps in the n, t network:

Feature maps of the components of the n and the t are not entirely independent from each others since they are representing the same (moving) object in the 3D space. The theorem 5.1 addresses this constraint.

Theorem 5.1. *Let f and g be two feature maps of size $H \times W$ are chosen from components of the n, t.*

$$f, g \in \{n_1, n_2, n_3, t_1, t_2, t_3\}. \tag{5.35}$$

Then the gradient vector of the f and the g are parallel with each other at each spatial point.

See Fig. 5.13. In this figure, there is a flexible disk in the center of the scene. The level set of the object on the feature map f and the feature map g are the borders of the differential width stripes and so they define the level sets of the two feature maps which are parallel to each other. So

$$\nabla f \times \nabla g = \vec{0}\ ,\ \forall\ f, g \in \{n_1, n_2, n_3, t_1, t_2, t_3\}. \tag{5.36}$$

Here $\times$ represent vector cross product. In practice, the synchrony between them are implemented in another way.

$$\nabla f \times \nabla g = 0 \longrightarrow f_x\ g_y - f_y\ g_x = 0 \tag{5.37}$$

This means

$$\frac{f_x}{f_y} = \frac{g_x}{g_y}\ ,\ \forall\ f, g \in \{n_1, n_2, n_3, t_1, t_2, t_3\}. \tag{5.38}$$

As a result

$$\frac{n_{1x}}{n_{1y}} = \frac{n_{2x}}{n_{2y}} = \frac{n_{3x}}{n_{3y}} = \frac{t_{1x}}{n_{1y}} = \frac{t_{2x}}{n_{2y}} = \frac{t_{3x}}{n_{3y}} \tag{5.39}$$

It means if one defines

$$V := [n_1, n_2, n_3, t_1, t_2, t_3]^T \tag{5.40}$$

then V_x and V_y are parallel. So the cos similarity of these two vector must be ± 1.

$$|cossim(V_x, V_y)| - 1 \tag{5.41}$$

As a result, the loss term can be defined as

$$\ell(n, t) = \ell(|cossim(V_x, V_y)| - 1) \tag{5.42}$$

The L1 measure is used as the ℓ.

Average length of t:

The difference between the average of the length of the ts over all the integer grid Ω (5.19) and 0.01 is punished using L1 norm.

5.4.3 Models of the depth, the optical flow and the n,t networks

Models which are used in the depth CNN and the optical flow and the n, t networks are depicted in Fig. 5.14. In case of the optical flow and the n, t networks, they are fed with the concatenated frames I_k and the I_t. Look at Fig. 5.14. This model benefits from the regularization effect of the spatial size for a better generalization. See chapter 3 for more details.

5.5 Experiments and results

The numerical experiments are conducted on the KITTI dataset [86] to evaluate the effectiveness of the proposed algorithm in comparison with the state-of-the-art methods. Also, the ablation studies are performed to better understand the contribution of the different settings of the designed cost function and the model.

5.5.1 Datasets

KITTI. The KITTI dataset contains over 93K outdoor images and the corresponding depth maps with an approximate resolution of 1240×374. All images are captured on driving cars by stereo cameras and a Lidar. The trained models are tested on 697 images from 29 scenes split by Eigen et al. [67]. All the images from the scenes in which one of them is in the test scenes are removed and the remaining RGB images and corresponding ground truth are used to train the models.

5.5.2 Implementation details

The overall training algorithm is depicted in Fig. 5.4. The depth network has the structure which is shown in Fig. 5.14 with some modifications. The number of channels in the first row of modules are 256, $C = 256$, and the rest of the modules has 512 channels, $C = 512$. In the encoder the Ns are equal to 4, $N = 4$ and in the decoder the Ns are equal to 1, $N = 1$. The other details are exactly similar to what was used in chapter 3. The optical flow network and the n, t network in Fig. 5.14 are exactly similar to the depth network except they are fed with concatenated the I_k and the I_t frames.

Other than the loss terms was described in 5.4.2, two other terms are used during optimization process. The first one punishes the depth map values out of $[d_{min},\ d_{max}]$. Here d_{min} is the minimum depth value and the d_{max} is the maximum depth value of the dataset used. The second one punishes the depth maps scale so that they do not grow large since there is no other mechanism to control the absolute value of the scale of the predicted depth map in the optimization. The L1 measure is used for these terms. The weights of these terms are shown in table 5.1.

The data augmentation is conducted on the training samples using the following methods. The RGB images and the corresponding depth maps are randomly resized with ratio [1, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8], randomly flipped horizontally, and finally randomly

cropped to 385×513. Note that the depth map should be scaled to the corresponding resizing ratio [67] in case of the supervised training which does not matter here.

In all of the experiments the base learning rate 1e-5 is used along with a learning rate scheduling going from 1 to 0 linearly for all training procedures on the KITTI dataset. The momentum for batch normalization is 0.01 and the batch size is 2. The weight decay and momentum in the SGD optimization are set to 0.0005 and 0.9 respectively. The model is trained for 555300 iterations on the KITTI dataset.

5.5.3 Evaluation metrics

Similar to [126] the performance is evaluated quantitatively based on mean absolute relative error (AbsRel), squared relative difference (SqRel), root mean squared error (RMS), root mean squared log of error (RMSlog) and the accuracy under threshold ($\sigma_i < 1.25^i \ , \ i = 1, 2, 3$). See section 1.5 for detailed formula of each measure.

5.5.4 Comparison with state-of-the-art

A comparison of our results with the state-of-the-art methods is shown in table 5.2 for the KITTI dataset. As shown in Table 5.2, our suggested method does not achieves comparable results in comparison with the state-of-the-art methods in the current setting. The reasons is discussed in section 5.6.

5.5.5 Ablation study

In this section, the effectiveness of the two cost function terms, the ”moving object detection loss” and the ”synchrony loss” are tested. All the implementation details are just like 5.5.2 except the total number of iterations is limited to 100 iterations. The comparison of the base model without the two terms and the complete algorithm based on Fig. 5.4 are shown in table 5.3. As the table shows the suggested loss terms are effective. See table 5.3.

5.6 Discussion and conclusion

The main benefit of this method is its ability to explicitly model flexible object and the entire dynamic scene without any assumptions on the number of moving objects. This emanates from relaxing geometric computer vision PnP and epipolar geometry from one motion model for the entire scene into one for every pixel in the scene. To change the optimization from under-constrained to over-constrained while keeping the flexibility of the model, the ”moving object detection loss” and the ”synchrony loss” were designed. The algorithm is trained in an unsupervised fashion. However, the whole algorithm comes with a big disadvantage. It is computationally expensive.

The training procedure is very slow that makes it difficult to compare with the state of the art. Also, it is not very stable which emanates from opencv's feature extraction. It needs to be scrutinized to find the main reason behind this instability, like testing the modules separately to make sure each is doing what is intended. Also, RAFT model [285] used as optical flow which was more difficult to train since the RAFT model is computationally more expansive than the original model from Fig. 5.14 and it is a very low flexible model. The model which is based on Fig. 5.14 is way more flexible and easier to train.

The backbone of most of the methods is based on two view-geometry. However three view geometry can deal with the occlusion better [92]. See Fig. 4.8. There are some tools to deal with three view geometry like three focal tensor. Last but not least, using Bayesian inference method [23], which is robust to noise as well as its ability to estimate the number of motion models makes it a good candidate as the filtering tool as well. Also changing the algorithm to a dynamic scene bundle adjustment setting is a natural extension on top the current algorithm. These are the future directions to extend this work.

Table 5.1: The weights of each loss term designed in the section 5.4.2.

The loss term	The corresponding weight
The sparse PnP loss	1.0
The dense PnP loss	1.0
The sparse epipolar loss	0.01
The dense epipolar loss	0.01
The Sparse optical flow loss	0.1
The dense optical flow loss	0.1
The moving object detection loss	1.0
The synchrony loss	1.0
The average size of the t	0.1
The depth scale loss	1e-4
The out of range depth loss	1.0

Table 5.2: Results on KITTI dataset as compared with state-of-the-art methods. The best result in each column (measure) is depicted in bold text. The second best is underlined. See section 1.5 for detailed formula of each measure.

Method	Error(lower is better)				Accuracy(higher is better)		
	AbsRel	SqRel	RMS	RMSlog	σ_1	σ_2	σ_3
Zhou et al. [361]	0.208	1.768	6.856	0.283	0.678	0.885	0.957
Yin et al. [348]	0.164	1.303	6.090	0.247	0.765	0.919	0.968
Mahjourian et al. [205]	0.163	1.240	6.220	0.250	0.762	0.916	0.968
Yin et al. [326]	0.155	1.296	5.857	0.233	0.793	0.931	0.973
Wang et al. [308]	0.151	1.257	5.583	0.228	0.810	0.936	0.974
Li et al. [182]	0.150	1.127	5.564	0.229	0.823	0.936	0.974
Zou et al. [366]	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Pilzer et al. [238]	0.142	1.231	5.785	0.239	0.795	0.924	0.968
Luo et al. [62]	0.141	1.029	5.350	0.216	0.816	0.941	0.976
Casser et al. [34]	0.141	1.026	5.291	0.215	0.816	0.945	0.979
Ranjan et al. [248]	0.140	1.070	5.326	0.217	0.826	0.941	0.975
Bian et al. [22]	0.137	1.089	5.439	0.217	0.830	0.942	0.975
Chen et al. [43]	0.135	1.070	5.230	0.210	0.841	0.948	0.980
Li et al. [178]	0.130	0.950	5.138	0.209	0.843	0.948	0.978
Gordon et al. [94]	0.128	0.959	5.230	0.212	0.845	0.947	0.976
Tosi et al. [295]	0.126	0.835	4.937	0.199	0.844	0.953	0.982
Godard et al. [92]	0.115	0.882	4.701	0.190	0.879	0.961	0.982
Guizilini et al. [96]	0.111	0.785	4.601	0.189	0.878	0.960	0.982
Guizilini et al. [96] (velocity sup.)	0.111	0.829	4.788	0.199	0.864	0.954	0.980
Johnston et al. [136]	0.111	0.941	4.817	0.189	0.885	0.961	0.981
Poggi et al. Boot+Self [241]	0.111	0.826	4.667	0.184	0.880	0.961	0.983
Poggi et al. Snap+Log [241]	0.117	0.900	4.838	0.192	0.873	0.958	0.981
Lee et al. [170]	0.112	0.777	4.772	0.191	0.872	0.959	0.982
Gao et al. [81]	0.112	0.866	4.693	0.189	0.881	0.961	0.981
Hui et al. [126]	0.108	0.710	4.513	0.183	0.884	0.964	0.983
Ours	0.495	0.532	10.551	0.524	0.324	0.612	0.809

Figure 5.1: A dynamic scene. Comparison between frame 12 and 16 of the ally 1 of the Sintel dataset. The camera is moving which is why the background of the scene (the static part of the scene) seems moving. The woman in the scene is the flexible moving object with respect to the scene. One motion model which describes the camera motion (the back ground or the static scene) is not enogh to model the scene.

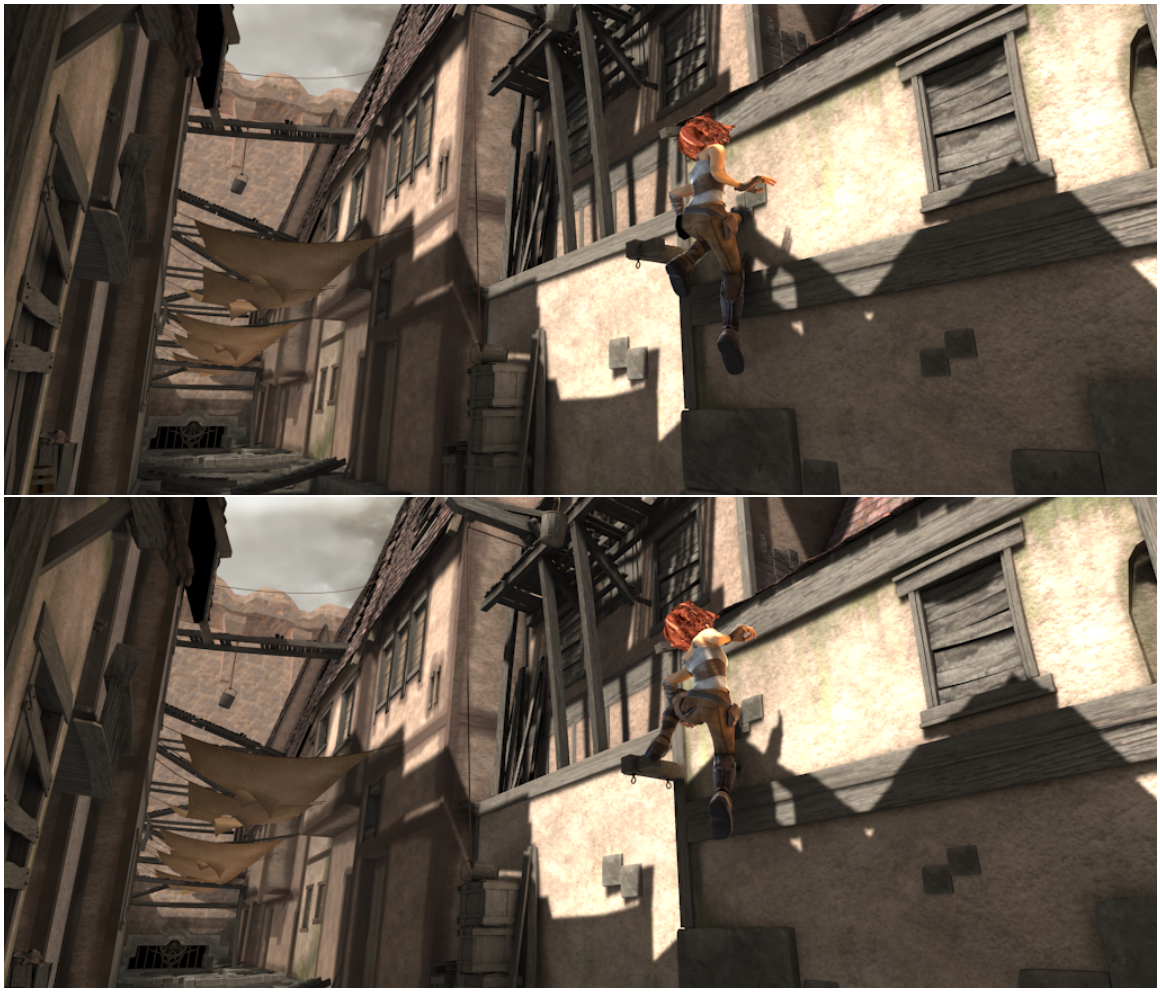
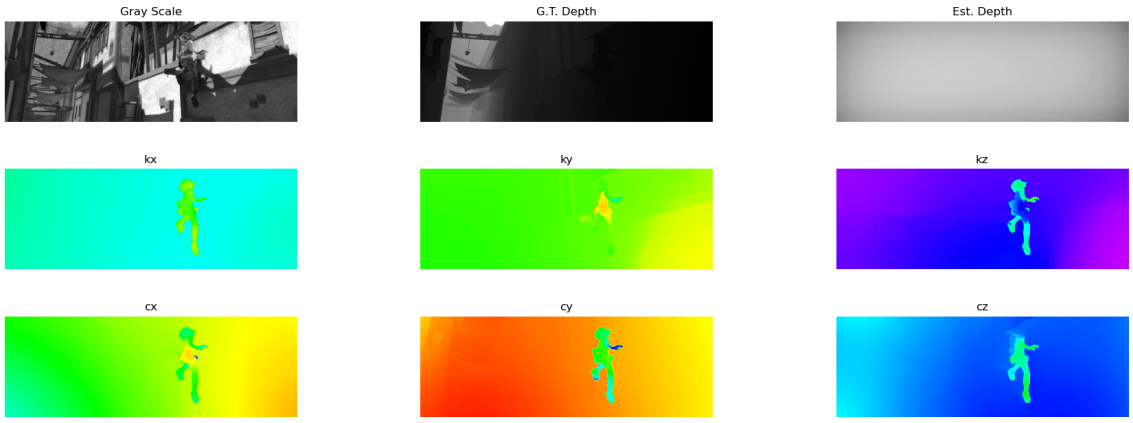
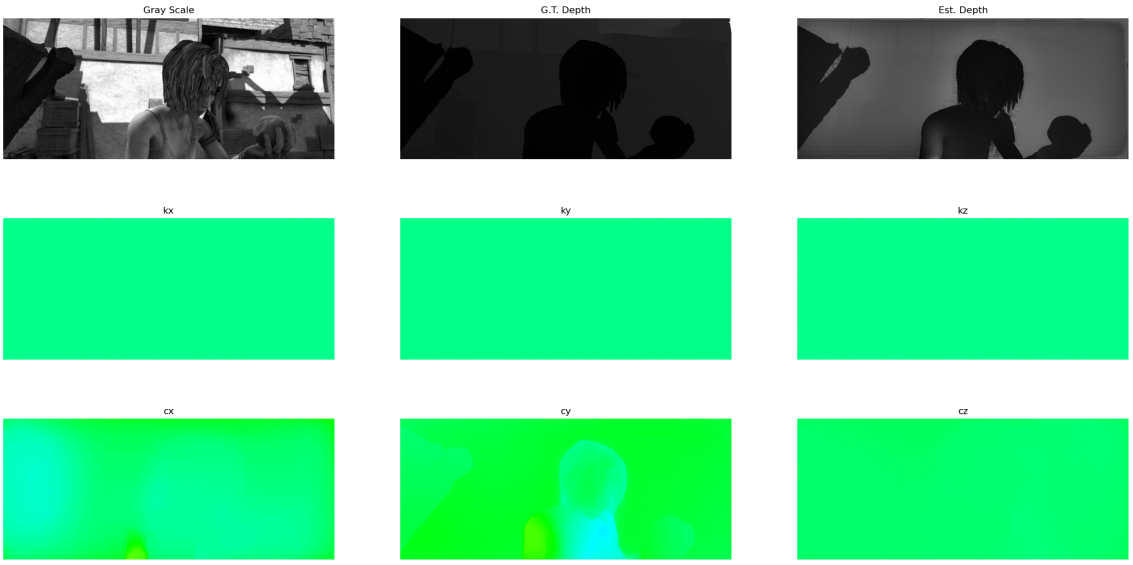


Figure 5.2: A dynamic scene. Comparison between frame 36 and 38 of the ally 2 of the Sintel dataset. The camera is moving which is why the background of the scene (the static part of the scene) seems moving. The woman in the scene is the flexible moving object with respect to the scene. One motion model which describes the camera motion (the back ground or the static scene) is not enough to model the scene.



(a)



(b)

Figure 5.3: Two examples, (a) and (b), of the flexible model for two frames structure from motion optimized on Sintel dataset. The input to this optimization is two consecutive RGB frames, a target frame and the key frame. The target frame is the gray-scale shown in top left gray-scale image. Other than these two images the ground truth opticalflow has been used as well. In the final plan, the opticalflow are extracted from the two RGB target and key frames using a CNN. In this method, I consider one motion model for every pixel in the scene instead of only one for the entire scene. The optimization utilizes two novel constrains, the "moving object detection loss" and the "synchrony loss" on the relaxed flexible scene model to be able to change the optimization from under constrained optimization to over constrained optimization. k_x, k_y and k_z are the three component of n and t_x, t_y and t_z are the three component of t. The details are discussed in 5.4.

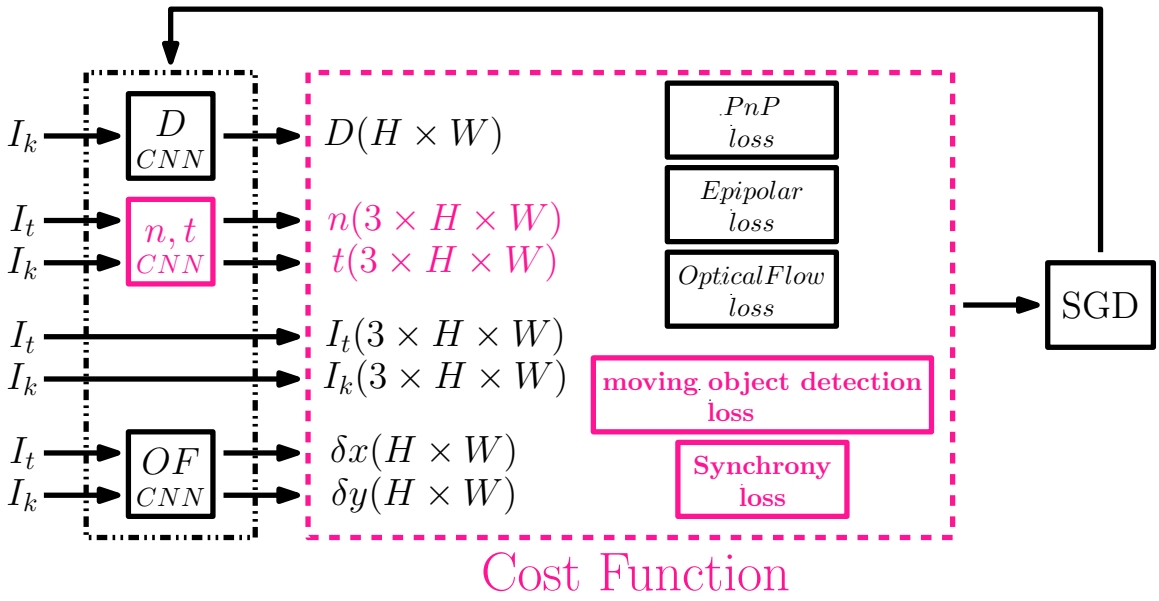


Figure 5.4: The overall training algorithm of MDE with explicit modeling of fully flexible objects and dynamic scene in an unsupervised manner. The inputs to the algorithm are the two consecutive RGB target and key frames, I_t and I_k. The depth network, the n and t network in pink box, and the optical-flow network are all CNN networks to estimate n and t and the optical-flow from the I_t, I_k frames and the depth map from the I_k frame. The estimated depth is of size $H \times W$, the n and t are each of size $3 \times H \times W$ and the optical-flow outputs are the change of location of pixels in x, y directions in RGB I_t, I_k images (δx, δy) are of size $H \times W$ each. The cost function is in blue box. The main idea in the paper is to relax the motion models to pixel level. The moving object detection loss and the Synchrony loss are providing the optimization with appropriate constraints so that the optimization become over constrained instead of being under constrained. The innovations of this paper are depicted in pink. The algorithm can easily be extended to three frames or even longer sequences of RGB frames.

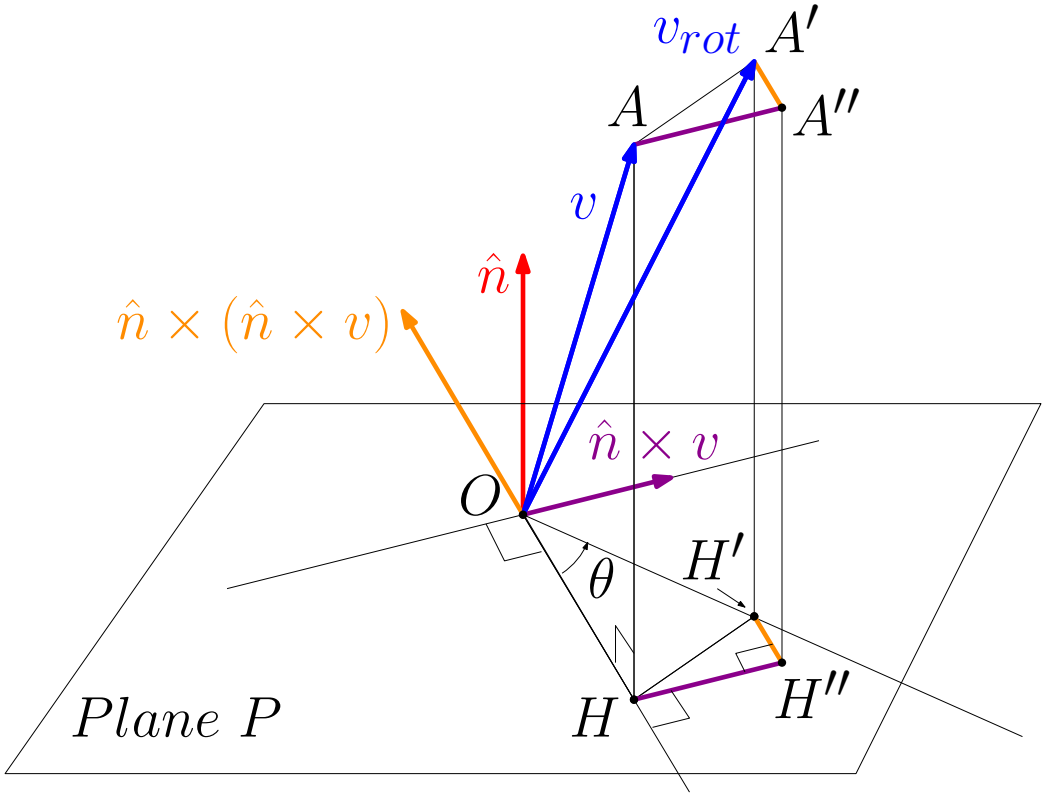


Figure 5.5: Rotation of a vector v around a unit vector $\hat{n}$ in 3D

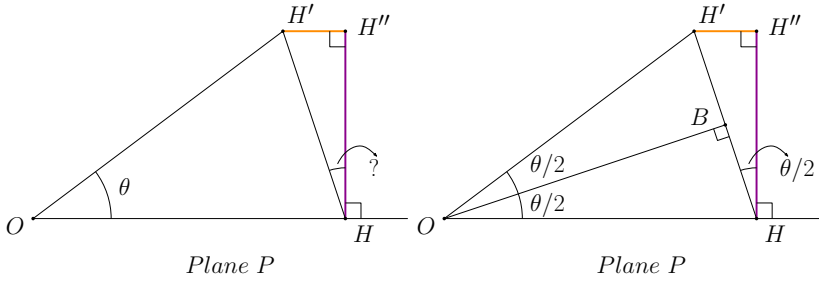


Figure 5.6: Projection into plane P from Fig. 5.5. Since the rotation happens around the vector n then $\|OH\| = \|OH'\|$. So the vector OB which is the bisector of the angle HOH' is perpendicular to HH'. Then the angle $H'HH$" is equal to $\theta/2$. Here $\|n\| = \theta$

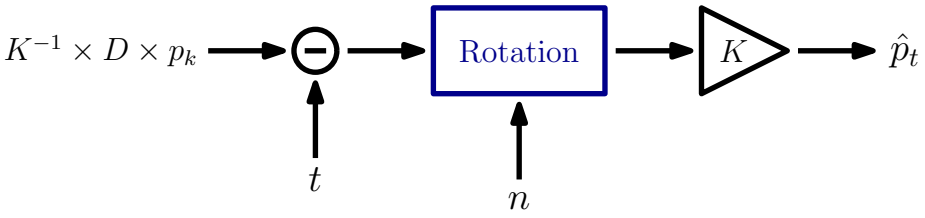


Figure 5.7: The block diagram of implementation of (5.6). The error is $cossim(p_t, \hat{p}_t) - 1$ where p_t comes from either the ground truth in AKAZE or from $p_k + [\delta x, \delta y, 0]^T$ in the dense case.

Figure 5.8: AKAZE features matched between two consecutive frames from NYUDV2.

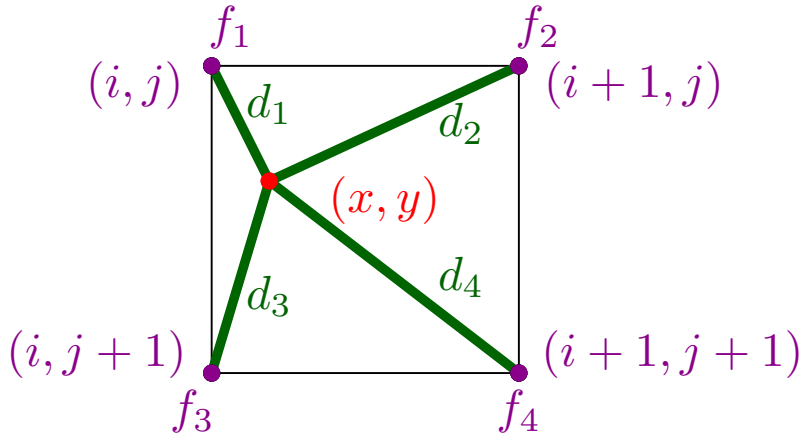


Figure 5.9: The output of the CNNs are available in integer grids Ω. However, AKAZE feature matching algorithm matched the points on the two frames I_t, I_k at arbitrary floating points (x, y).

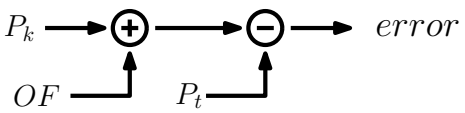


Figure 5.10: Sparse optical flow error

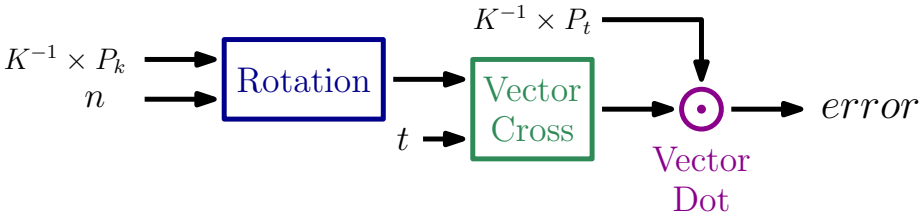


Figure 5.11: Dense epipolar implementation based on (5.3).

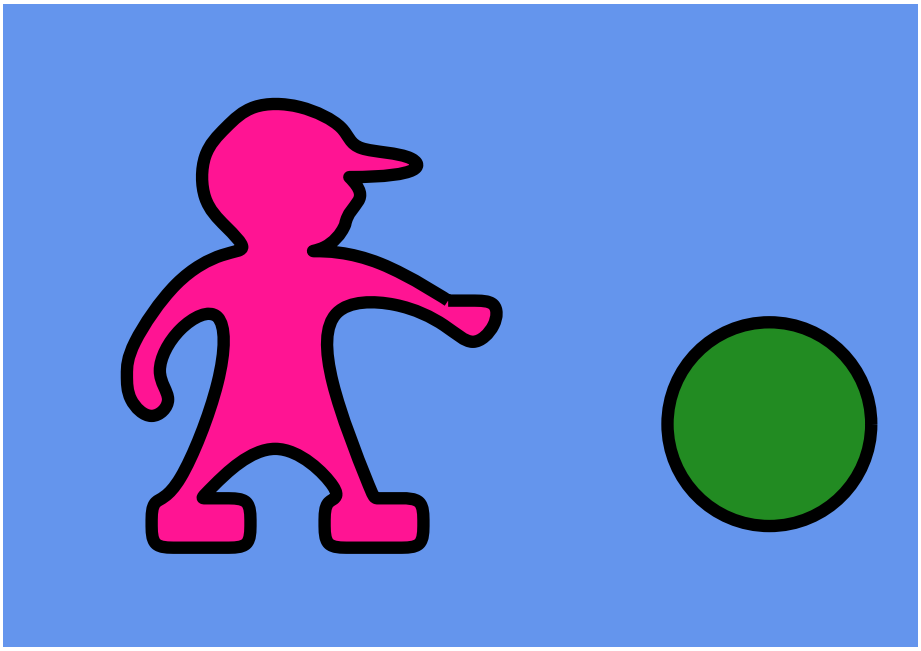


Figure 5.12: Result of enforcing motion model detection loss on a hypothetical dynamic scene. There are three sections in the scene. The blue part is the static part of the scene. A kid and a ball are the moving objects in the scene in pink and green respectively. Mask here is the black part of the scene which separates different areas from each others. The mask contains the extreme gradients values.

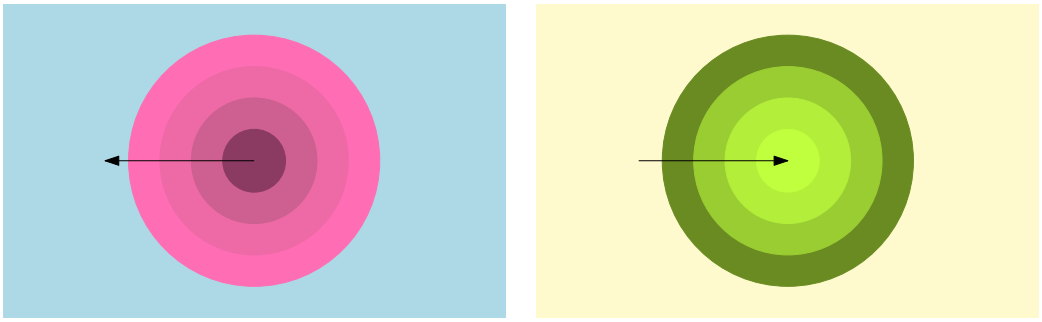


Figure 5.13: Two feature maps f, g from the set $\{n_1, n_2, n_3, t_1, t_2, t_3\}$. The corresponding scene is a dynamic scene which contains a flexible object. The round central shades represent the moving object. It is a flexible object since it has different shades as a result of having different values in the feature map. The background value represent the ego motion in static scene. The level sets between these two feature maps are the same if there is a change in the motion model in the scene in that area. This can be expressed as (5.36) and (5.39).

Table 5.3: The effectiveness of the two cost function terms, the "moving object detection loss" and the "synchrony loss" are tested. All the implementation details are just like 5.5.2 except the total number of iterations is limited to 100 iterations. The comparison of the base model without the two terms and the complete algorithm based on Fig. 5.4 are shown here. As the table shows the suggested loss terms are effective.

Method	Error(lower is better)				Accuracy(higher is better)		
	AbsRel	SqRel	RMS	RMSlog	σ_1	σ_2	σ_3
Base model	0.630	0.656	12.013	0.620	0.235	0.457	0.674
Base + Synchrony	0.595	0.569	11.642	0.592	0.237	0.472	0.710
Base + Motion det	0.594	0.570	11.468	0.592	0.242	0.477	0.709

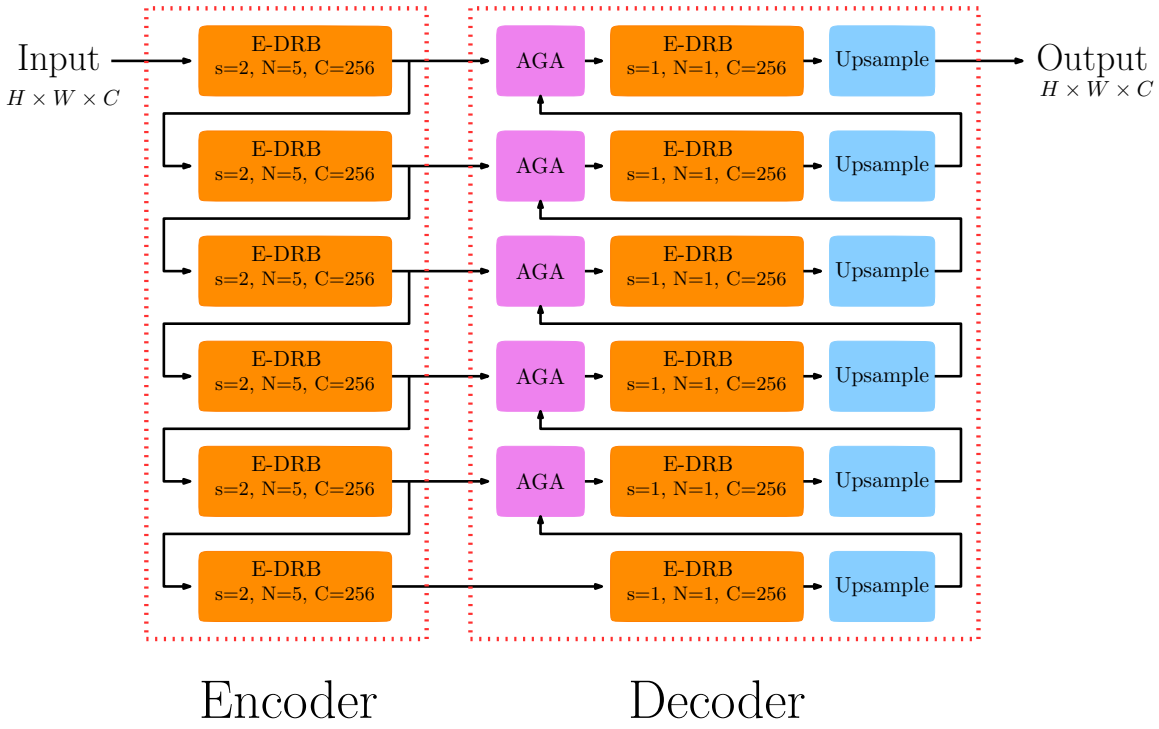


Figure 5.14: Models which are used in the depth CNN and the optical flow and n, t networks. In case of the optical flow and the n, t networks, they are fed with the concatenated frames I_k and the I_t. This model benefits from the regularization effect of the spatial size for a better generalization. See chapter 3 for more details.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

This study was conducted using two different paradigms. Single-image MDE, which was a supervised learning-based method, and the other was a multi-frame MDE in dynamic scene which contains flexible objects which was an unsupervised learning-based algorithm.

6.1.1 Single image MDE

The aim behind [224] was to exploit the similarities between the RGB picture and the corresponding depth map in the vicinity of geometric edges in the 3D environment. In other words, it is desirable to direct the encoder to generate better RGB features using the depth map in order to enhance the depth estimation quality at each spatial point. Nonetheless, the depth map is unavailable during the test phase. So the ultimate criterion for this guidance is the sensitivity-enhanced absolute value of cosine similarity between the local embedded features at each spatial point of the encoder and the decoder. It is permitted because the decoder's features are close to the the cost function during the training phase. The advantage of applying absolute value of local cosine similarity in embedded space over standard attention techniques, such as dot product, is that it is absolute and normalized, hence imposing tighter constraints on the network to better govern solution space. It is also

local, thus it does not complicate the optimization like the non-local versions.

It is important to note that for designing the suggested AGA module which uses the guidance of the depth map features to shape the RGB features, one might be able to assign more time and hardware resources to find more effective complex operations instead of $f_1(\mathcal{SA}_1) + f_2(\mathcal{SA}_2) \times \mathcal{CA}$ in Fig. 3.4 and (3.2). However, fine tuning the structure and parameters of such a module would be difficult. Hence, it was decided to use the divide-and-conquer strategy, where the guidance is divided into additive and multiplicative spatial attention weights, $f_1(\mathcal{SA}_1)$ and $f_2(\mathcal{SA}_2)$, and channel-wise attention weights $\mathcal{CA}$.

The suggested AGA module is light as all the computations are done locally and it adds only 0.03% (3e-4) of the total parameters of the base model to the model. So it can be added to any dense feature extraction module which has an encoder-decoder structure.

6.1.2 MDE in dynamic scene which contains flexible objects

MDE techniques in dynamic scenes rely heavily on either robustness or the estimation of a certain number of rigid motion models in the scene. The number of motion models is deemed fixed or must also be estimated. Moreover, these methods are not particularly designed to model dynamic scenes with flexible objects.

The primary objective of this paper was to explicitly model flexible objects and the full dynamic scene without making any assumptions about the number of moving objects in the scene. This is made achievable by relaxing geometric computer vision PnP and epipolar geometry restrictions from a single motion model for a portion of the scene to a single motion model for every pixel in the scene. This allows the model to detect even tiny, flexible, free-floating trash in a dynamic scene.

However, it makes the optimization under-constrained. To change the optimization from

under-constrained to over-constrained while maintaining the model's flexibility, "moving object detection loss" and "synchrony loss" are designed. Both loss terms impose restrictions on the components of the 3D rotation and translation vectors at each pixel and its 3×3 adjacent pixels. The algorithm is trained in an unsupervised fashion.

An ablation study is done to compare the extension of the computer vision paradigm from rigid to flexible bodies with and without the two loss terms. They demonstrate the usefulness of the "moving object detection loss" in detecting flexible moving objects within the scene, as well as the capabilities of the "synchrony loss" in syncing all six components of the 3D rotation and translation vectors, as they form the same scene.

The main results, nonetheless, are in no way comparable to the current state of the art. Due to the sluggish nature of the training procedure, it is difficult to compare it to the present state of the art. Additionally, the algorithm is unstable. Moreover, the optical flow model is incredibly noisy and naïve. To accelerate the process, internal second order optimization within the cost function must be added to the available stochastic gradient descent. In addition, the loss terms must be studied so that the main cause of this instability can be identified, such as by evaluating each module separately to confirm that it is operating as planned. As with the optical flow model, this algorithm requires a network with a high capacity. It is necessary to develop a novel optical flow network that meets the requirements of this algorithm.

In addition to the aforementioned drawbacks, the entire method has a significant disadvantage in comparison to the single image supervised MDE. Due to the lack of ground-truth data required to train the model, training the method is computationally expensive.

6.2 Future works

After completing the procedure in chapter 5, there are a few potential next paths that come to mind.

- Adapting the method to handle the interposition/occlusion: The majority of methods are founded on two-view geometry. However three view geometry can deal with the occlusion better [92]. See Fig. 4.8. In addition, there exist methods for modeling three-view geometry, such as the three focus tensor. To handle the interposition/occlusion challenge explicitly, the first step is to combine the two elements.

- Extending the algorithm to bundle adjustment in a dynamic scene containig flexible objects.

- Extending the algorithm to stereo bundle adjustment in a dynamic scene containing flexible objects to benefit from both temporal and stereo information.

- Adding loop closure

Last but not least, the Bayesian inference technique [23], which is noise-resistant and able to estimate the number of motion models, is a strong contender for the post-processing tool.

Bibliography

[1] Saddam Abdulwahab, Hatem A Rashwan, Miguel Angel Garcia, Mohammed Jabreel, Sylvie Chambon, and Domenec Puig. Adversarial learning for depth and viewpoint estimation from a single image. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9):2947–2958, 2020. 25, 26

[2] Shubhra Aich, Jean Marie Uwabeza Vianney, Md Amirul Islam, Mannat Kaur, and Bingbing Liu. Bidirectional attention network for monocular depth estimation. *arXiv preprint arXiv:2009.00743*, 2020. 46

[3] Vincent Aidala and Sherry Hammel. Utilization of modified polar coordinates for bearings-only tracking. *IEEE Transactions on automatic control*, 28(3):283–294, 1983. 92

[4] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected papers of hirotugu akaike*, pages 199–213. Springer, 1998. 87

[5] Ijaz Akhter, Yaser Sheikh, Sohaib Khan, and Takeo Kanade. Nonrigid structure from motion in trajectory space. *Advances in neural information processing systems*, 21, 2008. 95

[6] Pablo F Alcantarilla and T Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011. 106

[7] Pablo F Alcantarilla, José J Yebes, Javier Almazán, and Luis M Bergasa. On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments. In *2012 IEEE International Conference on Robotics and Automation*, pages 1290–1297. IEEE, 2012. 72, 75

[8] Filippo Aleotti, Fabio Tosi, Matteo Poggi, and Stefano Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 29, 39

[9] Ibraheem Alhashim and Peter Wonka. High quality monocular depth estimation via transfer learning. *arXiv preprint arXiv:1812.11941*, 2018. 22

[10] Yasin Almalioglu, Muhamad Risqi U Saputra, Pedro PB de Gusmao, Andrew Markham, and Niki Trigoni. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *2019 International conference on robotics and automation (ICRA)*, pages 5474–5480. IEEE, 2019. 29, 30

[11] Ali Jahani Amiri, Shing Yan Loo, and Hong Zhang. Semi-supervised monocular depth estimation with left-right consistency using deep neural network. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 602–607. IEEE, 2019. 30, 31

[12] Iro Armeni, Sasha Sax, Amir R Zamir, and Silvio Savarese. Joint 2d-3d-semantic data for indoor scene understanding. *arXiv preprint arXiv:1702.01105*, 2017. 16

[13] Amir Atapour-Abarghouei and Toby P Breckon. Real-time monocular depth estimation using synthetic data with domain adaptation via image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2800–2810, 2018. 32, 33

[14] Shai Avidan and Amnon Shashua. Trajectory triangulation of lines: Reconstruction of a 3d point moving along a line from a monocular image sequence. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 2, pages 62–66. IEEE, 1999. 91

[15] Shai Avidan and Amnon Shashua. Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(4):348–357, 2000. 91

[16] Mohammadreza Babaee, Duc Tung Dinh, and Gerhard Rigoll. A deep convolutional neural network for background subtraction. *arXiv preprint arXiv:1702.01731*, 2017. 74

[17] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Multi-view depth estimation by fusing single-view depth probability with multi-view geometry. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2842–2851, 2022. 85, 107

[18] Harry Barrow, J Tenenbaum, A Hanson, and E Riseman. Recovering intrinsic scene characteristics. *Comput. Vis. Syst*, 2(3-26):2, 1978. 18

[19] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008. 78

[20] Paul A Beardsley, Andrew Zisserman, and David William Murray. Navigation using affine structure from motion. In *European Conference on Computer Vision*, pages 85–96. Springer, 1994. 79

[21] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021. 23, 40

[22] Jiawang Bian, Zhichao Li, Naiyan Wang, Huangying Zhan, Chunhua Shen, Ming-Ming Cheng, and Ian Reid. Unsupervised scale-consistent depth and ego-motion learning from monocular video. *Advances in neural information processing systems*, 32, 2019. 29, 124

[23] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006. 122, 137

[24] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008. 71

[25] Terrance E Boult and L Gottesfeld Brown. Factorization-based segmentation of motions. In *Proceedings of the IEEE workshop on visual motion*, pages 179–180. IEEE Computer Society, 1991. 71

[26] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 2, pages 690–696. IEEE, 2000. 94, 108

[27] Michael D Breitenstein, Fabian Reichlin, Bastian Leibe, Esther Koller-Meier, and Luc Van Gool. Online multiperson tracking-by-detection from a single, uncalibrated camera. *IEEE transactions on pattern analysis and machine intelligence*, 33(9):1820–1833, 2010. 74

[28] Andrew Burton and John Radford. *Thinking in perspective: critical essays in the study of thought processes*, volume 646. Routledge, 1978. 75

[29] Arunkumar Byravan and Dieter Fox. Se3-nets: Learning rigid body motion using deep neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 173–180. IEEE, 2017. 89

[30] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 16

[31] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010. 78

[32] Ang Cao, Chris Rockwell, and Justin Johnson. Fwd: Real-time novel view synthesis with forward warping and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15713–15724, 2022. 84

[33] Yuanzhouhan Cao, Zifeng Wu, and Chunhua Shen. Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11):3174–3182, 2017. 24, 25, 39, 44, 45, 51

[34] Vincent Casser, Soeren Pirk, Reza Mahjourian, and Anelia Angelova. Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 8001–8008, 2019. 89, 108, 124

[35] Robert O Castle, Georg Klein, and David W Murray. Wide-area augmented reality using camera tracking and mapping in multiple regions. *Computer Vision and Image Understanding*, 115(6):854–867, 2011. 72

[36] Ayan Chakrabarti, Jingyu Shao, and Gregory Shakhnarovich. Depth from a single image by harmonizing overcomplete local network predictions. *arXiv preprint arXiv:1605.07081*, 2016. 67

[37] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 16

[38] Stephen M Chaves, Ayoung Kim, and Ryan M Eustice. Opportunistic sampling-based planning for active visual slam. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3073–3080. IEEE, 2014. 72

[39] Jinhui Chen and Jian Yang. Robust subspace segmentation via low-rank representation. *IEEE transactions on cybernetics*, 44(8):1432–1445, 2013. 89

[40] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 47

[41] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. *Advances in neural information processing systems*, 29, 2016. 16

[42] Xiaotian Chen, Xuejin Chen, and Zheng-Jun Zha. Structure-aware residual pyramid network for monocular depth estimation. *arXiv preprint arXiv:1907.06023*, 2019. 22, 39

[43] Yuhua Chen, Cordelia Schmid, and Cristian Sminchisescu. Self-supervised learning with geometric constraints in monocular video: Connecting flow, depth, and camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7063–7072, 2019. 124

[44] Yuru Chen, Haitao Zhao, Zhengwei Hu, and Jingchao Peng. Attention-based context aggregation network for monocular depth estimation. *International Journal of Machine Learning and Cybernetics*, 12(6):1583–1596, 2021. 24, 45

[45] Xinjing Cheng, Peng Wang, and Ruigang Yang. Depth estimation via affinity learned with convolutional spatial propagation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 103–119, 2018. 39

[46] Falak Chhaya, Dinesh Reddy, Sarthak Upadhyay, Visesh Chari, M Zeeshan Zia, and K Madhava Krishna. Monocular reconstruction of vehicles: Combining slam with

shape priors. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5758–5765. IEEE, 2016. 74

[47] Jaehoon Cho, Dongbo Min, Youngjung Kim, and Kwanghoon Sohn. A large rgb-d dataset for semi-supervised monocular depth estimation. *arXiv preprint arXiv:1904.10230*, 2019. 16, 30

[48] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 220–226. IEEE, 2005. 79

[49] Burcu Cinaz and Holger Kenn. Headslam-simultaneous localization and mapping with head-mounted inertial and laser range sensors. In *2008 12th IEEE International Symposium on Wearable Computers*, pages 3–10. IEEE, 2008. 71

[50] Colah. Understanding lstm. 40, 41

[51] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 16

[52] Joao Paulo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998. 71, 88, 93, 108

[53] Arun CS Kumar, Suchendra M Bhandarkar, and Mukta Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 283–291, 2018. 24

[54] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008. 80

[55] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 16

[56] Qi Dai, Vaishakh Patil, Simon Hecker, Dengxin Dai, Luc Van Gool, and Konrad Schindler. Self-supervised object motion and depth estimation from video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1004–1005, 2020. 28

[57] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6):1052–1067, 2007. 36

[58] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 52

[59] Maxime Derome, Aurelien Plyer, Martial Sanfourche, and Guy Le Besnerais. Moving object detection in real-time using stereo from a mobile platform. *Unmanned Systems*, 3(04):253–266, 2015. 73, 75

[60] Maxime Derome, Aurélien Plyer, Martial Sanfourche, and Guy Le Besnerais. Real-time mobile object detection using stereo. In *2014 13th International Conference on Control Automation Robotics & Vision (ICARCV)*, pages 1021–1026. IEEE, 2014. 75

[61] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016. 81

[62] Mingyu Ding, Zhe Wang, Bolei Zhou, Jianping Shi, Zhiwu Lu, and Ping Luo. Every frame counts: Joint learning of video segmentation and optical flow. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 10713–10720, 2020. 124

[63] Xingshuai Dong, Matthew A Garratt, Sreenatha G Anavatti, and Hussein A Abbass. Towards real-time monocular depth estimation for robotics: A survey. *arXiv preprint arXiv:2111.08600*, 2021. 16, 39, 41

[64] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 23

[65] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 76, 81

[66] David Eigen and Rob 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*, pages 2650–2658, 2015. 25, 26, 39, 44, 45, 67

[67] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014. 9, 21, 32, 39, 52, 53, 67, 120, 121

[68] Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013. 89

[69] Jakob Engel, Thomas Schöps, and Daniel Cremers. Lsd-slam: Large-scale direct monocular slam. In *European conference on computer vision*, pages 834–849. Springer, 2014. 83

[70] Zhicheng Fang, Xiaoran Chen, Yuhua Chen, and Luc Van Gool. Towards good practice for cnn-based monocular depth estimation. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1091–1100, 2020. 61, 67, 68

[71] Xiaohan Fei, Alex Wong, and Stefano Soatto. Geo-supervised visual depth prediction. *IEEE Robotics and Automation Letters*, 4(2):1661–1668, 2019. 27

[72] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 73, 79, 86

[73] Michael Fonder and Marc Van Droogenbroeck. Mid-air: A multi-modal dataset for extremely low altitude drone flights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 16

[74] Dariush Forouher, Marvin Große Besselmann, and Erik Maehle. Sensor fusion of depth camera and ultrasound data for obstacle detection and robot navigation. In *2016 14th international conference on control, automation, robotics and vision (ICARCV)*, pages 1–6. IEEE, 2016. 1

[75] Katerina Fragkiadaki, Pablo Arbelaez, Panna Felsen, and Jitendra Malik. Learning to segment moving objects in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4083–4090, 2015. 77

[76] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018. 23, 44, 51, 52, 61, 67, 68

[77] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial intelligence review*, 43(1):55–81, 2015. 80

[78] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 16

[79] Dorian Gálvez-López and Juan D Tardos. Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, 2012. 80

[80] Yukang Gan, Xiangyu Xu, Wenxiu Sun, and Liang Lin. Monocular depth estimation with affinity, vertical pooling, and label enhancement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 224–239, 2018. 44

[81] Feng Gao, Jincheng Yu, Hao Shen, Yu Wang, and Huazhong Yang. Attentional separation-and-aggregation network for self-supervised depth-pose learning in dynamic scenes. In *Conference on Robot Learning*, pages 2195–2205. PMLR, 2021. 124

[82] Xiao-Shan Gao, Xiao-Rong Hou, Jianliang Tang, and Hang-Fei Cheng. Complete solution classification for the perspective-three-point problem. *IEEE transactions on pattern analysis and machine intelligence*, 25(8):930–943, 2003. 79

[83] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, 2016. 27

[84] C William Gear. Multibody grouping from motion images. *International Journal of Computer Vision*, 29(2):133–150, 1998. 88, 89, 93

[85] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 16, 21, 22

[86] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 7, 51, 119

[87] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 963–968. Ieee, 2011. 82

[88] Arturo Gil, Óscar Reinoso, Mónica Ballesta, and Miguel Juliá. Multi-robot visual slam using a rao-blackwellized particle filter. *Robotics and Autonomous Systems*, 58(1):68–80, 2010. 72

[89] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 759–768, 2015. 76

[90] Susanna Gladh, Martin Danelljan, Fahad Shahbaz Khan, and Michael Felsberg. Deep motion features for visual tracking. In *2016 23rd international conference on pattern recognition (ICPR)*, pages 1243–1248. IEEE, 2016. 76

[91] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 270–279, 2017. 27, 31, 32

[92] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019. 4, 101, 122, 124, 137

[93] Alvina Goh and René Vidal. Segmenting motions of different types by unsupervised manifold clustering. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2007. 89

[94] Ariel Gordon, Hanhan Li, Rico Jonschkowski, and Anelia 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*, pages 8977–8986, 2019. 124

[95] H-M Gross, H-J Boehme, Christof Schröter, Steffen Müller, Alexander König, Ch Martin, Matthias Merten, and Andreas Bley. Shopbot: Progress in developing an interactive mobile shopping assistant for everyday use. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 3471–3478. IEEE, 2008. 72

[96] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2485–2494, 2020. 27, 124

[97] Vitor Guizilini, Rareș Ambruș, Dian Chen, Sergey Zakharov, and Adrien Gaidon. Multi-frame self-supervised depth with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 160–170, 2022. 84, 107

[98] Vitor Guizilini, Rui Hou, Jie Li, Rares Ambrus, and Adrien Gaidon. Semantically-guided representation learning for self-supervised monocular depth. *arXiv preprint arXiv:2002.12319*, 2020. 27, 45

[99] Vitor Guizilini, Jie Li, Rares Ambrus, Sudeep Pillai, and Adrien Gaidon. Robust semi-supervised monocular depth estimation with reprojected distances. In *Conference on robot learning*, pages 503–512. PMLR, 2020. 30, 31, 107

[100] Xiaoyan Guo and Wen Zheng. Improving monocular depth estimation by leveraging structural awareness and complementary datasets. *ECCV, Lecture Notes in Computer Science*, 2020. 46

[101] Xiaoyang Guo, Hongsheng Li, Shuai Yi, Jimmy Ren, and Xiaogang Wang. Learning monocular depth by distilling cross-domain stereo networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–500, 2018. 32

[102] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016. 76

[103] Akhil Gurram, Onay Urfalioglu, Ibrahim Halfaoui, Fahd Bouzaraa, and Antonio M López. Monocular depth estimation by learning from heterogeneous datasets. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 2176–2181. IEEE, 2018. 26

[104] Hyowon Ha, Sunghoon Im, Jaesik Park, Hae-Gon Jeon, and In So Kweon. High-quality depth from uncalibrated small motion clip. In *Proceedings of the IEEE conference on computer vision and pattern Recognition*, pages 5413–5421, 2016. 34

[105] Ankur Handa, Michael Bloesch, Viorica Pătrăucean, Simon Stent, John McCormac, and Andrew Davison. gvnn: Neural network library for geometric computer vision. In *European Conference on Computer Vision*, pages 67–82. Springer, 2016. 76

[106] Zhixiang Hao, Yu Li, Shaodi You, and Feng Lu. Detail preserving depth estimation from a single image using attention guided networks. In *2018 International Conference on 3D Vision (3DV)*, pages 304–313. IEEE, 2018. 46

[107] Liu Haomin, Zhang Guofeng, and Bao Hujun. A survey of monocular simultaneous localization and mapping. *Journal of Computer-Aided Design & Computer Graphics*, 28(6):855–868, 2016. 35

[108] Chris Harris and Carl Stennett. Rapid-a video rate object tracker. In *BMVC*, pages 1–6, 1990. 74

[109] Chris Harris, Mike Stephens, et al. A combined corner and edge detector. In *Alvey vision conference*, volume 15, pages 10–5244. Citeseer, 1988. 78

[110] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 74, 77

[111] Richard Hartley and Frederik Schaffalitzky. Powerfactorization: 3d reconstruction with missing or uncertain data. In *Australia-Japan advanced workshop on computer vision*, volume 74, pages 76–85, 2003. 93

[112] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003. 73, 87

[113] Richard I Hartley and Peter Sturm. Triangulation. *Computer vision and image understanding*, 68(2):146–157, 1997. 79, 91

[114] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 21

[115] Lei He, Jiwen Lu, Guanghui Wang, Shiyu Song, and Jie Zhou. Sosd-net: Joint semantic object segmentation and depth estimation from monocular images. *Neurocomputing*, 440:251–263, 2021. 45

[116] Minhyeok Heo, Jaehan Lee, Kyung-Rae Kim, Han-Ul Kim, and Chang-Su Kim. Monocular depth estimation using whole strip masking and reliability-based refinement. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 36–51, 2018. 39

[117] Stephan Heuel and Wolfgang Forstner. Matching, reconstructing and grouping 3d lines from multiple views using uncertain projective geometry. In *Proceedings*

of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, volume 2, pages II–II. IEEE, 2001. 74

[118] Berthold KP Horn. *Shape from shading: A method for obtaining the shape of a smooth opaque object from one view*. MIT, 1970. 18, 38

[119] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981. 75

[120] Stefan Hrabar, Gaurav S Sukhatme, Peter Corke, Kane Usher, and Jonathan Roberts. Combined optic-flow and stereo-based navigation of urban canyons for a uav. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3309–3316. IEEE, 2005. 72

[121] Yi-Yu Hsieh, Wei-Yu Lin, Dong-Lin Li, and Jen-Hui Chuang. Deep learning-based obstacle detection and depth estimation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1635–1639. IEEE, 2019. 26

[122] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 21, 48

[123] Junjie Hu, Mete Ozay, Yan Zhang, and Takayuki Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019. 22, 39, 44, 67

[124] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 6, 21

[125] Jingwei Huang, Zhili Chen, Duygu Ceylan, and Hailin Jin. 6-dof vr videos with a single 360-camera. In *2017 IEEE Virtual Reality (VR)*, pages 37–44. IEEE, 2017. 1

[126] Tak-Wai Hui. Rm-depth: Unsupervised learning of recurrent monocular depth in dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1675–1684, 2022. 83, 107, 121, 124

[127] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *European Conference on Computer Vision*, pages 581–597. Springer, 2020. 46, 67

[128] Naoyuki Ichimura. Motion segmentation based on factorization method and discriminant criterion. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 1, pages 600–605. IEEE, 1999. 93

[129] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017. 76

[130] Omid Hosseini Jafari, Oliver Groth, Alexander Kirillov, Michael Ying Yang, and Carsten Rother. Analyzing modular cnn architectures for joint depth prediction and semantic segmentation. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4620–4627. IEEE, 2017. 26

[131] Hossein Javidnia and Peter Corcoran. Accurate depth map estimation from small motions. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2453–2461, 2017. 34

[132] Rongrong Ji, Ke Li, Yan Wang, Xiaoshuai Sun, Feng Guo, Xiaowei Guo, Yongjian Wu, Feiyue Huang, and Jiebo Luo. Semi-supervised adversarial monocular depth estimation. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2410–2422, 2019. 30, 31

[133] Jianbo Jiao, Ying Cao, Yibing Song, and Rynson Lau. Look deeper into depth: Monocular depth estimation with semantic booster and attention-driven loss. In

Proceedings of the European conference on computer vision (ECCV), pages 53–69, 2018. 45, 46

[134] Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. Geometric structure based and regularized depth estimation from 360 indoor imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 889–898, 2020. 16

[135] Yongcheng Jing, Yezhou Yang, Zunlei Feng, Jingwen Ye, Yizhou Yu, and Mingli Song. Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*, 26(11):3365–3385, 2019. 33

[136] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 4756–4765, 2020. 28, 45, 46, 124

[137] Eagle S Jones and Stefano Soatto. Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4):407–430, 2011. 73

[138] Jae-Il Jung and Yo-Sung Ho. Depth map estimation from single-view image using object classification based on bayesian learning. In *2010 3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video*, pages 1–4. IEEE, 2010. 19

[139] Jeremy Yirmeyahu Kaminski and Mina Teicher. General trajectory triangulation. In *European Conference on Computer Vision*, pages 823–836. Springer, 2002. 91

[140] Jeremy Yirmeyahu Kaminski and Mina Teicher. A general framework for trajectory triangulation. *Journal of Mathematical Imaging and Vision*, 21(1):27–41, 2004. 91

[141] Ken-ichi Kanatani. Motion segmentation by subspace separation and model selection. In *Proceedings Eighth IEEE International Conference on computer Vision. ICCV 2001*, volume 2, pages 586–591. IEEE, 2001. 87, 88, 93

[142] Kenichi Kanatani. *Statistical optimization for geometric computation: theory and practice*. Courier Corporation, 2005. 87

[143] Kenichi Kanatani and Chikara Matsunaga. Estimating the number of independent motions for multibody motion segmentation. In *Asian Conference on Computer Vision*, pages 7–12. Citeseer, 2002. 86

[144] Kevin Karsch, Ce Liu, and Sing Bing Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE transactions on pattern analysis and machine intelligence*, 36(11):2144–2158, 2014. 18, 19

[145] Jinkyu Kim, Anna Rohrbach, Trevor Darrell, John Canny, and Zeynep Akata. Textual explanations for self-driving vehicles. In *Proceedings of the European conference on computer vision (ECCV)*, pages 563–578, 2018. 45

[146] Jens Klappstein, Tobi Vaudrey, Clemens Rabe, Andreas Wedel, and Reinhard Klette. Moving object segmentation using optical flow and depth information. In *Pacific-Rim Symposium on Image and Video Technology*, pages 611–623. Springer, 2009. 73, 75

[147] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE, 2007. 79

[148] Georg Klein and David Murray. Parallel tracking and mapping on a camera phone. In *2009 8th IEEE International Symposium on Mixed and Augmented Reality*, pages 83–86. IEEE, 2009. 72, 78

[149] Marvin Klingner, Jan-Aike Termöhlen, Jonas Mikolajczyk, and Tim Fingscheidt. Self-supervised monocular depth estimation: Solving the dynamic object problem

by semantic guidance. In *European Conference on Computer Vision*, pages 582–600. Springer, 2020. 45

[150] Maria Klodt and Andrea Vedaldi. Supervising the new with the old: learning sfm from sfm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 698–713, 2018. 28

[151] Kishore Konda and Roland Memisevic. Unsupervised learning of depth and motion. *arXiv preprint arXiv:1312.3429*, 2013. 81

[152] Naejin Kong and Michael J Black. Intrinsic depth: Improving depth transfer with intrinsic images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3514–3522, 2015. 18, 38

[153] Johannes Kopf, Xuejian Rong, and Jia-Bin Huang. Robust consistent video depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1611–1621, 2021. 75, 107

[154] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 76, 77, 81

[155] Suryansh Kumar, Yuchao Dai, and Hongdong Li. Multi-body non-rigid structure-from-motion. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 148–156. IEEE, 2016. 95

[156] Varun Ravi Kumar, Marvin Klingner, Senthil Yogamani, Stefan Milz, Tim Fingscheidt, and Patrick Mader. Syndistnet: Self-supervised monocular fisheye camera distance estimation synergized with semantic segmentation for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 61–71, 2021. 45

[157] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. g 2 o: A general framework for graph optimization. In *2011 IEEE*

International Conference on Robotics and Automation, pages 3607–3613. IEEE, 2011. 80

[158] Abhijit Kundu, K Madhava Krishna, and CV Jawahar. Realtime multibody visual slam with a smoothly moving monocular camera. In *2011 International Conference on Computer Vision*, pages 2080–2087. IEEE, 2011. 92

[159] Abhijit Kundu, K Madhava Krishna, and Jayanthi Sivaswamy. Moving object detection by multi-view geometric techniques from a single camera mounted robot. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4306–4312. IEEE, 2009. 73, 74

[160] Yevhen Kuznietsov, Jorg Stuckler, and Bastian Leibe. Semi-supervised deep learning for monocular depth map prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6647–6655, 2017. 30, 31, 32

[161] Lubor Ladicky, Jianbo Shi, and Marc Pollefeys. Pulling things out of perspective. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 89–96, 2014. 19, 67

[162] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth international conference on 3D vision (3DV)*, pages 239–248. IEEE, 2016. 21, 22, 39, 53, 67, 80

[163] Quoc Le, Alexandre Karpenko, Jiquan Ngiam, and Andrew Ng. Ica with reconstruction cost for efficient overcomplete feature learning. *Advances in neural information processing systems*, 24, 2011. 77

[164] Quoc V Le. Building high-level features using large scale unsupervised learning. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8595–8598. IEEE, 2013. 77

[165] J-P Le Cadre and Olivier Trémois. Bearings-only tracking for maneuvering sources. *IEEE Transactions on Aerospace and Electronic Systems*, 34(1):179–193, 1998. 92

[166] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 76, 82

[167] Jae-Han Lee and Chang-Su Kim. Multi-loss rebalancing algorithm for monocular depth estimation. In *Proceedings of the 2020 European Conference on Computer Vision (ECCV), Glasgow, UK*, pages 23–28, 2020. 67

[168] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 22, 44, 61, 67, 68

[169] Kuan-Hui Lee, Jenq-Neng Hwang, Greg Okapal, and James Pitton. Driving recorder based on-road pedestrian tracking using visual slam and constrained multiple-kernel. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 2629–2635. IEEE, 2014. 74

[170] Seokju Lee, Sunghoon Im, Stephen Lin, and In So Kweon. Learning monocular depth in dynamic scenes via instance-aware projection consistency. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1863–1872, 2021. 124

[171] Zeyu Lei, Yan Wang, Zijian Li, and Junyao Yang. Attention based multilayer feature fusion convolutional neural network for unsupervised monocular depth estimation. *Neurocomputing*, 423:343–352, 2021. 6

[172] Titus Leistner, Radek Mackowiak, Lynton Ardizzone, Ullrich Köthe, and Carsten Rother. Towards multimodal depth estimation from light fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12953–12961, 2022. 82

[173] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011. 78

[174] Stefan Leutenegger, Paul Furgale, Vincent Rabaud, Margarita Chli, Kurt Konolige, and Roland Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. *Proceedings of Robotis Science and Systems (RSS) 2013*, 2013. 73

[175] Bo Li, Yuchao Dai, and Mingyi He. Monocular depth estimation with hierarchical fusion of dilated cnns and soft-weighted-sum inference. *Pattern Recognition*, 83:328–339, 2018. 24, 25

[176] Bo Li, Chunhua Shen, Yuchao Dai, Anton Van Den Hengel, and Mingyi He. Depth and surface normal estimation from monocular images using regression on deep features and hierarchical crfs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1119–1127, 2015. 39, 67

[177] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. *arXiv preprint arXiv:2010.16404*, 2020. 95, 107

[178] Hanhan Li, Ariel Gordon, Hang Zhao, Vincent Casser, and Anelia Angelova. Unsupervised monocular depth learning in dynamic scenes. In *Conference on Robot Learning*, pages 1908–1917. PMLR, 2021. 124

[179] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3372–3380, 2017. 44, 67

[180] Rui Li, Qing Mao, Pei Wang, Xiantuo He, Yu Zhu, Jinqiu Sun, and Yanning Zhang. Semantic-guided representation enhancement for self-supervised monocular trained depth estimation. *arXiv preprint arXiv:2012.08048*, 2020. 45

[181] Ruibo Li, Ke Xian, Chunhua Shen, Zhiguo Cao, Hao Lu, and Lingxiao Hang. Deep attention-based classification network for robust depth prediction. In *Asian Conference on Computer Vision*, pages 663–678. Springer, 2018. 24, 25, 47, 48

[182] Shunkai Li, Fei Xue, Xin Wang, Zike Yan, and Hongbin Zha. Sequential adversarial learning for self-supervised deep visual odometry. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2851–2860, 2019. 124

[183] Ting Li, Vinutha Kallem, Dheeraj Singaraju, and René Vidal. Projective factorization of multiple rigid-body motions. In *CVPR*, pages 1–6. Citeseer, 2007. 94

[184] Yuyan Li, Yuliang Guo, Zhixin Yan, Xinyu Huang, Ye Duan, and Liu Ren. Omnifusion: 360 monocular depth estimation via geometry-aware fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2801–2810, 2022. 82

[185] Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018. 16

[186] Lukas Liebel and Marco Körner. Multidepth: Single-image depth estimation via multi-task regression and classification. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1440–1447. IEEE, 2019. 25

[187] Hyon Lim, Jongwoo Lim, and H Jin Kim. Real-time 6-dof monocular visual slam in a large-scale environment. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 1532–1539. IEEE, 2014. 78, 79

[188] Tsung-Han Lin and Chieh-Chih Wang. Deep learning of spatio-temporal features with geometric-based moving point detection for motion segmentation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3058–3065. IEEE, 2014. 77

[189] Beyang Liu, Stephen Gould, and Daphne Koller. Single image depth estimation from predicted semantic labels. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1253–1260. IEEE, 2010. 19

[190] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5162–5170, 2015. 45, 67

[191] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian Reid. Learning depth from single monocular images using deep convolutional neural fields. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2024–2039, 2015. 21

[192] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics (ToG)*, 28(3):1–9, 2009. 1

[193] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):171–184, 2012. 89

[194] Jing Liu, Xiaona Zhang, Zhaoxin Li, and Tianlu Mao. Multi-scale residual pyramid attention network for monocular depth estimation. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 5137–5144. IEEE, 2021. 6

[195] Jun Liu, Qing Li, Rui Cao, Wenming Tang, and Guoping Qiu. Mininet: An extremely lightweight convolutional neural network for real-time unsupervised monocular depth estimation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:255–267, 2020. 30

[196] Miaomiao Liu, Mathieu Salzmann, and Xuming He. Discrete-continuous depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2014. 19, 67

[197] Peng Liu, Zonghua Zhang, Zhaozong Meng, and Nan Gao. Joint attention mechanisms for monocular depth estimation with multi-scale convolutions and adaptive weight adjustment. *IEEE Access*, 8:184437–184450, 2020. 61, 67, 68

[198] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 77

[199] H Christopher Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293(5828):133–135, 1981. 75, 79

[200] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 78

[201] Bruce D Lucas, Takeo Kanade, et al. *An iterative image registration technique with an application to stereo vision*, volume 81. Vancouver, 1981. 78

[202] Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM Transactions on Graphics (ToG)*, 39(4):71–1, 2020. 83

[203] Fangchang Ma, Guilherme Venturelli Cavalheiro, and Sertac Karaman. Self-supervised sparse-to-dense: Self-supervised depth completion from lidar and monocular camera. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3288–3295. IEEE, 2019. 27

[204] Fangchang Ma and Sertac Karaman. Sparse-to-dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 4796–4803. IEEE, 2018. 39

[205] Reza Mahjourian, Martin Wicke, and Anelia 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*, pages 5667–5675, 2018. 27, 124

[206] Michele Mancini, Gabriele Costante, Paolo Valigi, and Thomas A Ciarfuglia. Fast robust monocular depth estimation for obstacle detection with fully convolutional networks. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4296–4303. IEEE, 2016. 22

[207] Michele Mancini, Gabriele Costante, Paolo Valigi, and Thomas A Ciarfuglia. J-mod 2: Joint monocular obstacle detection and depth estimation. *IEEE Robotics and Automation Letters*, 3(3):1490–1497, 2018. 16

[208] Alina Marcu, Dragos Costea, Vlad Licaret, Mihai Pîrvu, Emil Slusanschi, and Marius Leordeanu. Safeuav: Learning to estimate depth and safe landing areas for uavs from synthetic data. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 16

[209] Alwyn Mathew, Aditya Prakash Patra, and Jimson Mathew. Self-attention dense depth estimation network for unrectified video sequences. *arXiv preprint arXiv:2005.14313*, 2020. 46

[210] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas 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*, pages 4040–4048, 2016. 28, 32, 76

[211] John McCormac, Ankur Handa, Stefan Leutenegger, and Andrew J Davison. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2678–2687, 2017. 16

[212] Ishit Mehta, Parikshit Sakurikar, and PJ Narayanan. Structured adversarial training for unsupervised monocular depth estimation. In *2018 International Conference on 3D Vision (3DV)*, pages 314–323. IEEE, 2018. 29, 30, 39

[213] Christopher Mei, Gabe Sibley, Mark Cummins, Paul Newman, and Ian Reid. Rslam: A system for large-scale mapping in constant-time using stereo. *International journal of computer vision*, 94(2):198–214, 2011. 71, 72

[214] Iaroslav Melekhov, Juha Ylioinas, Juho Kannala, and Esa Rahtu. Relative camera pose estimation using convolutional neural networks. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 675–687. Springer, 2017. 80, 81

[215] Davide Migliore, Roberto Rigamonti, Daniele Marzorati, Matteo Matteucci, and Domenico G Sorrenti. Use a single camera for simultaneous localization and mapping with mobile object tracking in dynamic environments. In *ICRA Workshop on Safe navigation in open and dynamic environments: Application to autonomous vehicles*, pages 12–17, 2009. 74

[216] Vikram Mohanty, Shubh Agrawal, Shaswat Datta, Arna Ghosh, Vishnu Dutt Sharma, and Debashish Chakravarty. Deepvo: A deep learning approach for monocular visual odometry. *arXiv preprint arXiv:1611.06069*, 2016. 81

[217] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Monocular vision based slam for mobile robots. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 1027–1031. IEEE, 2006. 71

[218] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Generic and real-time structure from motion. In *British Machine Vision Conference 2007 (BMVC 2007)*, 2007. 79

[219] Etienne Mouragnon, Maxime Lhuillier, Michel Dhome, Fabien Dekeyser, and Patrick Sayd. Generic and real-time structure from motion using local bundle adjustment. *Image and Vision Computing*, 27(8):1178–1193, 2009. 79, 80

[220] Peter Muller and Andreas Savakis. Flowdometry: An optical flow and deep learning based approach to visual odometry. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 624–631. IEEE, 2017. 81

[221] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 80

[222] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017. 28

[223] Yohei Murakami, Takeshi Endo, Yoshimichi Ito, and Noboru Babaguchi. Depth-estimation-free condition for projective factorization and its application to 3d reconstruction. In *Asian Conference on Computer Vision*, pages 150–162. Springer, 2012. 94

[224] Taher Naderi, Amir Sadovnik, Jason Hayward, and Hairong Qi. Monocular depth estimation with adaptive geometric attention. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 944–954, 2022. 7, 15, 43, 56, 58, 60, 61, 62, 63, 64, 65, 66, 67, 68, 70, 134

[225] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 7, 51, 52

[226] David Nistér. An efficient solution to the five-point relative pose problem. *IEEE transactions on pattern analysis and machine intelligence*, 26(6):756–770, 2004. 75, 79

[227] David Nistér, Oleg Naroditsky, and James Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. Ieee, 2004. 78

[228] Chaoyue Niu, Danesh Tarapore, and Klaus-Peter Zauner. Low-viewpoint forest depth dataset for sparse rover swarms. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 8035–8040. IEEE, 2020. 16

[229] Diego Ortin and José Marıa Martınez Montiel. Indoor robot motion based on monocular images. *Robotica*, 19(3):331–342, 2001. 88

[230] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. 93

[231] Kemal E Ozden, Konrad Schindler, and Luc Van Gool. Multibody structure-from-motion in practice. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1134–1141, 2010. 88

[232] Marco Paladini, Alessio Del Bue, Marko Stosic, Marija Dodig, Joao Xavier, and Lourdes Agapito. Factorization for non-rigid and articulated structure using metric projections. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2898–2905. IEEE, 2009. 95

[233] Jiahao Pang, Wenxiu Sun, Jimmy SJ Ren, Chengxi Yang, and Qiong Yan. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 887–895, 2017. 31

[234] Hyun Soo Park, Takaaki Shiratori, Iain Matthews, and Yaser Sheikh. 3d reconstruction of a moving point from a series of 2d projections. In *European conference on computer vision*, pages 158–171. Springer, 2010. 91

[235] Hyun Soo Park, Takaaki Shiratori, Iain Matthews, and Yaser Sheikh. 3d trajectory reconstruction under perspective projection. *International Journal of Computer Vision*, 115(2):115–135, 2015. 91

[236] Andra Petrovai and Sergiu Nedevschi. Exploiting pseudo labels in a self-supervised learning framework for improved monocular depth estimation. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1578–1588, 2022. 84, 107

[237] Massimo Piccardi. Background subtraction techniques: a review. In *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, volume 4, pages 3099–3104. IEEE, 2004. 74

[238] Andrea Pilzer, Stephane Lathuiliere, Nicu Sebe, and Elisa Ricci. Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9768–9777, 2019. 124

[239] Andrea Pilzer, Dan Xu, Mihai Puscas, Elisa Ricci, and Nicu Sebe. Unsupervised adversarial depth estimation using cycled generative networks. In *2018 International Conference on 3D Vision (3DV)*, pages 587–595. IEEE, 2018. 29

[240] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. Towards real-time unsupervised monocular depth estimation on cpu. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5848–5854. IEEE, 2018. 30

[241] Matteo Poggi, Filippo Aleotti, Fabio Tosi, and Stefano Mattoccia. On the uncertainty of self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3227–3237, 2020. 124

[242] Charan D Prakash, Jinjin Li, Farshad Akhbari, and Lina J Karam. Sparse depth calculation using real-time key-point detection and structure from motion for advanced driver assist systems. In *International Symposium on Visual Computing*, pages 740–751. Springer, 2014. 34

[243] Vignesh Prasad and Brojeshwar Bhowmick. Sfmlearner++: Learning monocular depth & ego-motion using meaningful geometric constraints. In *2019 IEEE Winter*

Conference on Applications of Computer Vision (WACV), pages 2087–2096. IEEE, 2019. 28

[244] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10318–10327, 2021. 96

[245] Xiaojuan Qi, Renjie Liao, Zhengzhe Liu, Raquel Urtasun, and Jiaya Jia. Geonet: Geometric neural network for joint depth and surface normal estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 283–291, 2018. 26, 44, 67

[246] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. *arXiv preprint arXiv:2103.13413*, 2021. 6

[247] Rene Ranftl, Vibhav Vineet, Qifeng Chen, and Vladlen Koltun. Dense monocular depth estimation in complex dynamic scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4058–4066, 2016. 4, 90, 106, 108

[248] Anurag Ranjan, Varun Jampani, Lukas Balles, Kihwan Kim, Deqing Sun, Jonas Wulff, and Michael 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*, pages 12240–12249, 2019. 29, 124

[249] Jouni Rantakokko, Joakim Rydell, Peter Strömbäck, Peter Händel, Jonas Callmer, David Törnqvist, Fredrik Gustafsson, Magnus Jobs, and Mathias Grudén. Accurate and reliable soldier and first responder indoor positioning: multisensor systems and cooperative localization. *IEEE Wireless Communications*, 18(2):10–18, 2011. 71

[250] Shankar Rao, Roberto Tron, Rene Vidal, and Yi Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2009. 89

[251] S Hussain Raza, Omar Javed, Aveek Das, Harpreet Sawhney, Hui Cheng, and Irfan Essa. Depth extraction from videos using geometric context and occlusion boundaries. *arXiv preprint arXiv:1510.07317*, 2015. 19

[252] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 26

[253] Jorma Rissanen. Universal coding, information, prediction, and estimation. *IEEE Transactions on Information theory*, 30(4):629–636, 1984. 87

[254] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 6

[255] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European conference on computer vision*, pages 430–443. Springer, 2006. 78

[256] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 80

[257] Reza Sabzevari and Davide Scaramuzza. Monocular simultaneous multi-body motion segmentation and reconstruction from perspective views. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 23–30. IEEE, 2014. 86, 88

[258] Reza Sabzevari and Davide Scaramuzza. Multi-body motion estimation from monocular vehicle-mounted cameras. *IEEE Transactions on Robotics*, 32(3):638–651, 2016. 76, 86, 88

[259] Dalila Sánchez-Escobedo, Xiao Lin, Josep R Casas, and Montse Pardas. Hybridnet for depth estimation and semantic segmentation. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1563–1567. IEEE, 2018. 25

[260] Muhamad Risqi U Saputra, Andrew Markham, and Niki Trigoni. Visual slam and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)*, 51(2):1–36, 2018. 72, 95, 97, 98, 99, 100

[261] Muhamad Risqi Utama Saputra, Paulus Insap Santosa, et al. Obstacle avoidance for visually impaired using auto-adaptive thresholding on kinect's depth image. In *2014 IEEE 11th Intl Conf on Ubiquitous Intelligence and Computing and 2014 IEEE 11th Intl Conf on Autonomic and Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, pages 337–342. IEEE, 2014. 72

[262] Ashutosh Saxena, Sung Chung, and Andrew Ng. Learning depth from single monocular images. *Advances in neural information processing systems*, 18, 2005. 19

[263] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008. 16

[264] Davide Scaramuzza. 1-point-ransac structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints. *International journal of computer vision*, 95(1):74–85, 2011. 76, 88

[265] Davide Scaramuzza, Friedrich Fraundorfer, and Roland Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point ransac. In *2009 IEEE International conference on robotics and automation*, pages 4293–4299. Ieee, 2009. 76, 88

[266] Konrad Schindler and David Suter. Two-view multibody structure-and-motion with outliers. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 643–648. IEEE, 2005. 87

[267] Konrad Schindler and David Suter. Two-view multibody structure-and-motion with outliers through model selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):983–995, 2006. 86

[268] Konrad Schindler, David Suter, and Hanzi Wang. A model-selection framework for multibody structure-and-motion of image sequences. *International Journal of Computer Vision*, 79(2):159–177, 2008. 87

[269] Konrad Schindler, Hanzi Wang, et al. Perspective n-view multibody structure-and-motion through model selection. In *European Conference on Computer Vision*, pages 606–619. Springer, 2006. 87

[270] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 78

[271] Thomas Schops, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017. 16

[272] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978. 87

[273] Amnon Shashua, Shai Avidan, and Michael Werman. Trajectory triangulation over conic section. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 330–336. IEEE, 1999. 91

[274] Gabe Sibley, Christopher Mei, Ian Reid, and Paul Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *The International Journal of Robotics Research*, 29(8):958–980, 2010. 71

[275] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 16, 21, 22

[276] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 76

[277] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 16

[278] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1746–1754, 2017. 16

[279] Wenfeng Song, Shuai Li, Ji Liu, Aimin Hao, Qinping Zhao, and Hong Qin. Contextualized cnn for scene-aware depth estimation from single rgb image. *IEEE Transactions on Multimedia*, 22(5):1220–1233, 2019. 25

[280] Andrew Spek, Thanuja Dharmasiri, and Tom Drummond. Cream: Condensed real-time models for depth prediction using convolutional neural networks. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 540–547. IEEE, 2018. 26

[281] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012. 16

[282] Peter Sturm and Bill Triggs. A factorization based algorithm for multi-image projective structure and motion. In *European conference on computer vision*, pages 709–720. Springer, 1996. 93

[283] Wen Su, Haifeng Zhang, Quan Zhou, Wenzhen Yang, and Zengfu Wang. Monocular depth estimation using information exchange network. *IEEE Transactions on Intelligent Transportation Systems*, 2020. 45, 61, 67, 68

[284] Wei Tan, Haomin Liu, Zilong Dong, Guofeng Zhang, and Hujun Bao. Robust monocular slam in dynamic environments. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 209–218. IEEE, 2013. 71, 74

[285] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 76, 99, 122

[286] Ninad Thakoor, Jean Gao, and Venkat Devarajan. Multibody structure-and-motion segmentation by branch-and-bound model selection. *IEEE Transactions on Image Processing*, 19(6):1393–1402, 2010. 86, 88

[287] Hu Tian and Fei Li. Semi-supervised depth estimation from a single image based on confidence learning. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8573–8577. IEEE, 2019. 30, 31

[288] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International journal of computer vision*, 9(2):137–154, 1992. 92, 94

[289] Alessio Tonioni, Matteo Poggi, Stefano Mattoccia, and Luigi Di Stefano. Unsupervised domain adaptation for depth prediction from images. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2396–2409, 2019. 32, 33

[290] Philip HS Torr. Geometric motion segmentation and model selection. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 356(1740):1321–1340, 1998. 80, 87, 108

[291] Philip HS Torr and Andrew Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and vision Computing*, 15(8):591–605, 1997. 79

[292] Philip HS Torr and Andrew Zisserman. Feature based methods for structure and motion estimation. In *International workshop on vision algorithms*, pages 278–294. Springer, 1999. 78

[293] Antonio Torralba and Aude Oliva. Depth estimation from image structure. *IEEE Transactions on pattern analysis and machine intelligence*, 24(9):1226–1238, 2002. 19

[294] Fabio Tosi, Filippo Aleotti, Matteo Poggi, and Stefano Mattoccia. Learning monocular depth estimation infusing traditional stereo knowledge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9799–9809, 2019. 27

[295] Fabio Tosi, Filippo Aleotti, Pierluigi Zama Ramirez, Matteo Poggi, Samuele Salti, Luigi Di Stefano, and Stefano Mattoccia. Distilled semantics for comprehensive scene understanding from videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4654–4665, 2020. 124

[296] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *International workshop on vision algorithms*, pages 298–372. Springer, 1999. 102

[297] Shimon Ullman. The interpretation of structure from motion. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 203(1153):405–426, 1979. 33

[298] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5038–5047, 2017. 108

[299] Julien Valentin, Adarsh Kowdle, Jonathan T Barron, Neal Wadhwa, Max Dzitsiuk, Michael Schoenberg, Vivek Verma, Ambrus Csaszar, Eric Turner, Ivan Dryanovski, et al. Depth from motion for smartphone ar. *ACM Transactions on Graphics (ToG)*, 37(6):1–19, 2018. 1

[300] Sepehr Valipour, Mennatullah Siam, Martin Jagersand, and Nilanjan Ray. Recurrent fully convolutional networks for video segmentation. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 29–36. IEEE, 2017. 77

[301] Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 16

[302] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 23, 45

[303] Ehsan Elhamifar René Vidal. Sparse subspace clustering. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 00*, volume 6, pages 2790–2797, 2009. 89

[304] René Vidal. Online clustering of moving hyperplanes. *Advances in Neural Information Processing Systems*, 19, 2006. 89

[305] Rene Vidal, Yi Ma, and Shankar Sastry. Generalized principal component analysis (gpca). *IEEE transactions on pattern analysis and machine intelligence*, 27(12):1945–1959, 2005. 89

[306] Sudheendra Vijayanarasimhan, Susanna Ricco, Cordelia Schmid, Rahul Sukthankar, and Katerina Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804*, 2017. 28, 80, 83, 89

[307] Anjie Wang, Zhijun Fang, Yongbin Gao, Songchao Tan, Shanshe Wang, Siwei Ma, and Jenq-Neng Hwang. Adversarial learning for joint optimization of depth and ego-motion. *IEEE Transactions on Image Processing*, 29:4130–4142, 2020. 30

[308] Chaoyang Wang, José Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2022–2030, 2018. 124

[309] Chaoyang Wang, Simon Lucey, Federico Perazzi, and Oliver Wang. Web stereo video supervision for depth prediction from dynamic scenes. In *2019 International Conference on 3D Vision (3DV)*, pages 348–357. IEEE, 2019. 108

[310] Chieh-Chih Wang, Charles Thorpe, Sebastian Thrun, Martial Hebert, and Hugh Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *The International Journal of Robotics Research*, 26(9):889–916, 2007. 71

[311] Chieh-Chih Wang and Chuck Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings 2002 IEEE International conference on robotics and automation (Cat. No. 02CH37292)*, volume 3, pages 2918–2924. IEEE, 2002. 71

[312] Dequan Wang, Coline Devin, Qi-Zhi Cai, Fisher Yu, and Trevor Darrell. Deep object-centric policies for autonomous driving. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8853–8859. IEEE, 2019. 45

[313] Fu-En Wang, Yu-Hsuan Yeh, Min Sun, Wei-Chen Chiu, and Yi-Hsuan Tsai. Bifuse: Monocular 360 depth estimation via bi-projection fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 462–471, 2020. 23

[314] Jianrong Wang, Ge Zhang, Mei Yu, Tianyi Xu, and Tao Luo. Attention-based dense decoding network for monocular depth estimation. *IEEE Access*, 8:85802–85812, 2020. 46, 61, 67, 68

[315] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 6

[316] Lijun Wang, Jianming Zhang, Oliver Wang, Zhe Lin, and Huchuan Lu. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 541–550, 2020. 45, 67

[317] Lijun Wang, Jianming Zhang, Yifan Wang, Huchuan Lu, and Xiang Ruan. Cliffnet for monocular depth estimation with hierarchical embedding loss. In *European Conference on Computer Vision*, pages 316–331. Springer, 2020. 67

[318] Linda Wang, Mahmoud Famouri, and Alexander Wong. Depthnet nano: A highly compact self-normalizing neural network for monocular depth estimation. *arXiv preprint arXiv:2004.08008*, 2020. 26

[319] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pages 2800–2809, 2015. 25, 26, 67

[320] Rui Wang, Stephen M Pizer, and Jan-Michael Frahm. Recurrent neural network for (un-) supervised learning of monocular video visual odometry and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5555–5564, 2019. 24

[321] Sen Wang, Ronald Clark, Hongkai Wen, and Niki Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2043–2050. IEEE, 2017. 80, 82

[322] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 45, 59

[323] Yin-Tien Wang, Ming-Chun Lin, and Rung-Chi Ju. Visual slam and moving-object detection for a small-size humanoid robot. *International Journal of Advanced Robotic Systems*, 7(2):13, 2010. 74

[324] Somkiat Wangsiripitak and David W Murray. Avoiding moving outliers in visual slam by tracking moving objects. In *2009 IEEE international conference on robotics and automation*, pages 375–380. IEEE, 2009. 74

[325] Oliver Wasenmüller, Marcel Meyer, and Didier Stricker. Corbs: Comprehensive rgb-d benchmark for slam using kinect v2. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–7. IEEE, 2016. 16

[326] Jamie Watson, Oisin Mac Aodha, Victor Prisacariu, Gabriel Brostow, and Michael Firman. The temporal opportunist: Self-supervised multi-frame monocular depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1164–1174, 2021. 124

[327] Wikipedia. Aerial perspective. 12, 14

[328] Wikipedia. Perspective graphical. 12, 13

[329] Diana Wofk, Fangchang Ma, Tien-Ju Yang, Sertac Karaman, and Vivienne Sze. Fastdepth: Fast monocular depth estimation on embedded systems. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6101–6108. IEEE, 2019. 26

[330] Changchang Wu. Towards linear-time incremental structure from motion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 127–134. IEEE, 2013. 78

[331] Changchang Wu, Sameer Agarwal, Brian Curless, and Steven M Seitz. Multicore bundle adjustment. In *CVPR 2011*, pages 3057–3064. IEEE, 2011. 79

[332] Shouchuan Wu, Haitao Zhao, and Shaoyuan Sun. Depth estimation from infrared video using local-feature-flow neural network. *International Journal of Machine Learning and Cybernetics*, 10(9):2563–2572, 2019. 16

[333] Jing Xiao, Jin-xiang Chai, and Takeo Kanade. A closed-form solution to non-rigid shape and motion recovery. In *European conference on computer vision*, pages 573–587. Springer, 2004. 95

[334] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 52

[335] Dan Xu, Elisa Ricci, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017. 45, 67

[336] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3917–3925, 2018. 45

[337] Feng Xue, Guirong Zhuo, Ziyuan Huang, Wufei Fu, Zhuoyue Wu, and Marcelo H Ang. Toward hierarchical self-supervised monocular absolute depth estimation for autonomous driving applications. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2330–2337. IEEE, 2020. 84

[338] Koichiro Yamanaka, Ryutaroh Matsumoto, Keita Takahashi, and Toshiaki Fujii. Adversarial patch attacks on monocular depth estimation networks. *IEEE Access*, 8:179094–179104, 2020. 7, 105

[339] Jingyu Yan and Marc Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *European conference on computer vision*, pages 94–106. Springer, 2006. 89

[340] Congyuan Yang, Daniel Robinson, and Rene Vidal. Sparse subspace clustering with missing entries. In *International Conference on Machine Learning*, pages 2463–2472. PMLR, 2015. 89

[341] Fengting Yang, Xiaolei Huang, and Zihan Zhou. Deep depth from focus with differential focus volume. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12642–12651, 2022. 85

[342] Gengshan Yang and Deva Ramanan. Upgrading optical flow to 3d scene flow through optical expansion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1334–1343, 2020. 3

[343] Gengshan Yang and Deva Ramanan. Learning to segment rigid motions from two frames. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1266–1275, 2021. 3, 90, 108

[344] Guanglei Yang, Hao Tang, Mingli Ding, Nicu Sebe, and Elisa Ricci. Transformers solve the limited receptive field for monocular depth prediction. *arXiv preprint arXiv:2103.12091*, 2021. 6, 67

[345] Guorun Yang, Xiao Song, Chaoqin Huang, Zhidong Deng, Jianping Shi, and Bolei Zhou. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 899–908, 2019. 16

[346] Xinchen Ye, Shude Chen, and Rui Xu. Dpnet: Detail-preserving network for high quality monocular depth estimation. *Pattern Recognition*, 109:107578, 2021. 45

[347] Wei Yin, Yifan Liu, Chunhua Shen, and Youliang Yan. Enforcing geometric constraints of virtual normal for depth prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5684–5693, 2019. 22, 44, 46, 47, 48, 50, 51, 52, 53, 54, 61, 64, 67, 68, 70

[348] Zhichao Yin and Jianping Shi. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1983–1992, 2018. 29, 124

[349] Georges Younes, Daniel Asmar, and Elie Shammas. A survey on non-filter-based monocular visual slam systems. *arXiv preprint arXiv:1607.00470*, 413:414, 2016. 80

[350] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 6

[351] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3916–3925, 2022. 46

[352] Min Yue, Guangyuan Fu, Ming Wu, and Hongqiao Wang. Semi-supervised monocular depth estimation based on semantic supervision. *Journal of Intelligent & Robotic Systems*, 100(2):455–463, 2020. 31

[353] Pierluigi Zama Ramirez, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi Di Stefano. Geometry meets semantics for semi-supervised monocular depth estimation. In *Asian Conference on Computer Vision*, pages 298–313. Springer, 2018. 30, 31

[354] Haokui Zhang, Chunhua Shen, Ying Li, Yuanzhouhan Cao, Yu Liu, and Youliang Yan. Exploiting temporal consistency for real-time video depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1725–1734, 2019. 24

[355] Teng Zhang, Arthur Szlam, and Gilad Lerman. Median k-flats for hybrid linear modeling with many outliers. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 234–241. IEEE, 2009. 89

[356] Yilun Zhang, Ty Nguyen, Ian D Miller, Shreyas S Shivakumar, Steven Chen, Camillo J Taylor, and Vijay Kumar. Dfinenet: Ego-motion estimation and depth refinement from sparse, noisy depth input with rgb guidance. *arXiv preprint arXiv:1903.06397*, 2019. 27

[357] Shanshan Zhao, Huan Fu, Mingming Gong, and Dacheng Tao. Geometry-aware symmetric domain adaptation for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9788–9798, 2019. 31, 33

[358] Wang Zhao, Shaohui Liu, Yezhi Shu, and Yong-Jin Liu. Towards better generalization: Joint depth-pose learning without posenet. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9151–9161, 2020. 29

[359] Zixiang Zhao, Jiangshe Zhang, Shuang Xu, Zudi Lin, and Hanspeter Pfister. Discrete cosine transform network for guided depth map super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5697–5707, 2022. 44

[360] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 767–783, 2018. 32, 33

[361] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1851–1858, 2017. 27, 28, 32, 80, 83, 124

[362] Wei Zhuo, Mathieu Salzmann, Xuming He, and Miaomiao Liu. Indoor scene structure analysis for single image depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 614–622, 2015. 67

[363] Nikolaos Zioulis, Antonis Karakottas, Dimitrios Zarpalas, and Petros Daras. Omnidepth: Dense depth estimation for indoors spherical panoramas. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 448–465, 2018. 16, 23

[364] Danping Zou and Ping Tan. Coslam: Collaborative visual slam in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence*, 35(2):354–366, 2012. 72, 74

[365] Hongwei Zou, Ke Xian, Jiaqi Yang, and Zhiguo Cao. Mean-variance loss for monocular depth estimation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1760–1764. IEEE, 2019. 24, 25

[366] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 36–53, 2018. 29, 107, 124

Vita

Taher Naderi completed his BS and MS degrees in Iran. His MS degree is in electrical engineering with a concentration in control systems from the university of Tehran. He finished his PhD in electrical engineering as well. His main research interest is 3D computer vision since he is curious why primates and birds are so successful in 3D perception of the scene and navigate through obstacles effortlessly.