Fakultät für Mathematik, Informatik und Naturwissenschaften
Lehr- und Forschungsgebiet Informatik VIII
Computer Vision
Prof. Dr. Bastian Leibe

Seminar Report

Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints

Nikolay Paleshnikov
Matriculation Number: 331623

June 2018

Advisor: Aljoša Ošep

Abstract

In this seminar, we discuss the most recent developement in the structure from motion research field incorporating unsupervised learning. After a theoretical problem formulation and the definition of an objective function used for estimation error optimization, we present a short summary of the past decade's progress in the training of deep neural networks, emphasizing on its implications for the structure from motion domain in particular. We proceed with a succint overview of the most prominent analytic approaches to depth and ego-motion estimation, as well as their learning-based counterparts. We also provide a comparison of the discussed methods in terms of estimation accuracy, operational robustness and applicability in a variety of environments. We finally delve into a detailed treatment of a newly developed deep network for depth and ego-motion estimation trained solely under self-supervision and present the evaluation results from the original publication.

Contents

1 Introduction

Visual odometry and Simultaneous Localization and Mapping (SLAM) are among the main topics of scientific research in the overlapping area of robotics and computer vision. Their applications range from virtual and augmented reality to unknown surface exploration and autonomous navigation. The problem settings have been clearly defined for a multitude of applications and environments, for which high-performing solutions on standardized evaluation datasets have been developed. Nonetheless, a general framework solving visual odometry and SLAM robustly under varying conditions and with a high-level geometry understanding is yet to be found. Open problems include life-long operation, resilience in a variety of environments and non-rigidity of the observed scene. At the same time, deep learning has greatly enhanced the notion of what can be learned by a neural network and has thus brought about a paradigm shift from analytic to statistical solutions in the computer science research. Self-supervised deep learning networks have already been successfully employed for structure and motion reconstruction, object detection and tracking, monocular depth estimation and optical flow prediction. This has given rise to the development of unsupervised learning approaches to monocular depth and ego-motion estimation circumventing most of the usual analytic steps in the solution in order to achieve enhanced estimation accuracy and robustness in a multitude of environments. The deep network proposed by Zhou et al. and trained under self-supervision has delivered competitive results in comparison with supervised or analytical frameworks. Its estimation precision has been outdone by the deep learning framework developed by Mahjourian et al. that costitutes the main focus of this report. The major benefit is due to the definition of a differentiable three-dimensional geometry loss term that substantially aids the objective function optimization. The superiority of the new method against established ones has been shown under multiple evaluation settings. Also worth pointing out is its competitive performance even in case of training the deep network on an unrelated custom dataset and subsequently evaluating it without fine-tuning on the test set.

This report is structured as follows. In Section 2, we present the preliminaries essential for the understanding of the subject at hand. We first introduce some basic projective geometry concepts necessary for the understanding of Structure from Motion (SfM) and bundle adjustment. We then elucidate the theoretical foundations underlying monocular visual odometry and monocular SLAM. As a next step, we present a concise outline of the development of deep learning in the context of computer vision. It is followed by a presentation of the most relevant related work in Section 3. A presentation of the discussed method can be found in Section 4. It commences with a short method overview, then provides a detailed description of the loss function used for optimization and focuses particularly on the main contribution of the proposed method, namely the definition of a novel three-dimensional geometry loss term. At its end, we also delineate the network architecture and the learning setup. The evaluation of the method against other state-of-the-art deep neural networks and SLAM frameworks is presented in Section 5. We conclude this report with a summary of the most important contributions of the discussed paper and some suggestions for further research in Section 6.

2 Preliminaries

Useful references for the theoretical foundations explained in this section are the excellent surveys of SfM [31] and SLAM [5]. For the more general geometric notions, consult [18].

2.1 Monocular Structure from Motion

A monocular SfM framework delivers a three-dimensional structure estimate of the observed environment based on the video sequence recorded by a single camera placed on a moving object. The various SfM techniques developed in the last few decades construct a geometric model of the moving camera and its environment and subsequently minimize a geometric reprojection error defined over sample points at different video frames. In the following, we formally define perspective point projection in three-dimensional space under the pinhole camera model and a simple objective function that can be used for bundle adjustment.

2.1.1 Point Projection and Reprojection

The transformation between any two frames in three-dimensional space can be represented as a member of the special Euclidean group $SE(3)$ under the rigid-body assumption. It can be formulated as a rotation $R \in \mathbb{R}^{3\times 3}$, followed by a translation $t \in \mathbb{R}^3$, and written in a single matrix as follows:

$$T = \begin{pmatrix} & R & & t \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

A point in three-dimensional space is represented in Euclidean coordinates augmented by a fourth component:

$$p = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \in \mathbb{R}^4$$

in order to facilitate the representation of an $SE(3)$ transformation by a single matrix multiplication $M_T : \mathbb{R}^4 \to \mathbb{R}^4$, $p \mapsto T\, p$.

The transformation matrix T can also be interpreted as a transformation of point coordinates from the local camera frame to the global world frame and therefore describes the camera pose at a given frame in world coordinates. The transformation from camera i with transformation matrix T_i to camera j with transformation matrix T_j is computed as:

$$T_i^j = T_j^{-1}\, T_i \tag{1}$$

The camera pose is thus described by the extrinsic camera parameters. Each camera also has intrinsic parameters such as the local coordinates of its principle point (c_x, c_y), where the image center lies, and the focal distances f_x and f_y, which are equal in a true pinhole camera model but can differ in case of non-square pixels. The intrinsic parameters constitute the camera matrix:

$$K = \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

which represents a transformation from three-dimensional space into the projective space of the camera and has been extended to comply with augmented point coordinates.

We use Euclidean coordinates for the representation of a pixel in a given camera frame, written in bold so as to prevent confusion with points in three-dimensional space:

$$\mathbf{p} = \begin{pmatrix} u \\ v \end{pmatrix} \in \mathbb{R}^2$$

The mapping of a point from the projective space of a camera into its image plane located at depth one can then be formulated as the map:

$$\Pi : \mathbb{R}^4 \to \mathbb{R}^2 \ , \ \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \mapsto \begin{pmatrix} x/z \\ y/z \end{pmatrix} \tag{2}$$

whereas its reverse is defined as:

$$\Pi^{-1} : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^4 \ , \ (\begin{pmatrix} u \\ v \end{pmatrix}, d_p) \mapsto \begin{pmatrix} d_p \cdot u \\ d_p \cdot v \\ d_p \\ 1 \end{pmatrix}$$

with d_p denoting the estimated depth of an image pixel $\mathbf{p}$.

We can already define the projection of a point p from augmented three-dimensional space to a point $\mathbf{p}$ lying in the image plane of camera i:

$$\mathbf{p} = \Pi \ (K \ p) \tag{3}$$

We can also define the reprojection of a point $\mathbf{p}$ lying in the image plane of camera i with estimated depth d_p to a point $\mathbf{p}'$ lying in the image plane of camera j as follows:

$$\mathbf{p}' = \Pi \ (K \ T_i^j \ K^{-1} \ \Pi^{-1} \ (\mathbf{p}, d_p)) \tag{4}$$

2.1.2 Bundle Adjustment

In a typical bundle adjustment setting, we are presented with a collection of images taken at consecutive time steps by a single camera. In each image, a set of sample points has been chosen and their correspondences to points in three-dimensional space has been established, e.g. by means of feature extraction and matching. The total geometric reprojection error can then be defined as a squared loss function:

$$E_{total} = \sum_{i \in F} \sum_{\mathbf{p}^* \in sp(i)} ||\mathbf{p}^* - \mathbf{p}||_2^2$$

where F stands for the set of frames, $sp(i)$ – for the set of sample points in frame i and $\mathbf{p}$ – for the projection of the point $p \in P$ from the set of scene points in three-dimensional space P corresponding to $\mathbf{p}^*$ as defined in the projection from Equation 3.

We can then formulate the objective function of a simple bundle adjustment instance as follows:

$$\underset{P, \{T_i | i \in F\}, K}{\text{argmin}} \ E_{total} \tag{5}$$

taking into account that all frames have been captured by a single camera with camera matrix K.

Bundle adjustment aims to deliver a Maximum a Posteriori (MAP) estimation of the frame poses and the point positions in the projective geometry model. In its simplest form, it is a non-linear non-convex least-squares optimization problem that often converges to a local minimum far from the global one in the absence of good initialization. In order to tackle this difficulty, the eight-point algorithm presented next can be employed for the computation of a good initialization.

2.1.3 Eight-Point Algorithm

First described in [24], the eight-point algorithm solves SfM for a pair of frames by fixing one of them as the origin of the coordinate system and thus only estimating the relative displacement of the other one.

Let us fix frame j so that we estimate the pose of the other frame i. It can be represented as an SE(3) transformation from camera to world coordinates, i.e. from the local coordinates in frame i to the local coordinates at frame j. This transformation consists of a rotation R and a translation t. Let us also use an abridged version of the camera matrix:

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 3}$$

We first state the following relation between a point $\mathbf{p}$ in the image plane at frame i and its corresponding point $\mathbf{p}'$ in the image plane at frame j:

$$K^{-1}\,\mathbf{p}' = R\,K^{-1}\,\mathbf{p} + t$$

By first taking the cross-product with t from the left, then preforming left multiplication with $K^{-1}\,\mathbf{p}'$ on both sides and finally leaving out all terms equal to zero, we obtain:

$$(K^{-1}\,\mathbf{p}')^T\, t \times R\,K^{-1}\,\mathbf{p} = 0$$

This relation can be written more succinctly as $\mathbf{p}'^T\, F\, \mathbf{p} = 0$ by defining the fundamental matrix F as $F = K^{-T}\,[t]_\times\, R\,K^{-1}$. Note that the skew-symmetric matrix $[t]_\times$ is used for the representation of the cross-product as a matrix multiplication.

Given eight point correspondences $(\mathbf{p}_k, \mathbf{p}'_k)_{k\in[1,\ldots,8]}$ between the images taken at frames i and j, we obtain a system of eight linear equations $\mathbf{p}'^T_k\, F\, \mathbf{p}_k = 0$, which can be solved for the entries of F by means of singular value decomposition. The rotation and translation are then extracted back from the fundamental matrix as explained in detail in [24].

By solving SfM for pairs of frames and subsequently triangulating points in space, we obtain a good initialization for bundle adjustment optimization. More efficient methods for the solution of the general SfM problem in a monocular setting are presented in the following subsections.

2.2 Monocular Visual Odometry and SLAM

Monocular visual odometry pertains to the ego-motion estimation based on the visual input from a single camera. It is usually represented as a piecewise linear trajectory connecting the three-dimensional pose estimates at selected keyframes from the input video. The pose of each new keyframe is estimated either as a displacement from the last keyframe or by means of reprojection error minimization in a continuously updated local optimization window of keyframes with a substantial amount of sample point correspondences.

When bundle adjustment is performed incrementally and in parallel with the visual odometry, as well as a map of the currently explored environment is being constructed, we are dealing with SLAM. The bundle adjustment optimization acts globally on all keyframes created until then and makes use of additional constraints such as loop closures. This ensures that places visited multiple times coincide on the map so that the actual topology of the environment is correctly represented.

2.2.1 Probabilistic Interpretation

The probabilistic technique employed in SLAM is a MAP estimation similar to the one done by means of bundle adjustment. We describe it according to the outline in [5]. The interdependence among the variables being optimized for SLAM can be visualized by the aid of a factor graph such as the one shown in Figure 1, where all optimization variables are drawn as circles. Let us wrap them up in a single unknown variable X that needs to be estimated. For each of its subsets $X_i \subseteq X$ consisting of variables connected to an unique factor i in the factor graph, there is a measurement $z_i \in Z$ related to the variables in X_i by a function $z_i = h_i(X_i) + \varepsilon_i$. h_i is an analytically derived, typically non-linear function fitting the observation model, i.e. projective geometry for visual SLAM, so that e.g. in the case of a point observation factor, h_i performs point projection. ε_i stands for measurement noise. The objective function can then be formulated as the maximization of the posterior probability or the belief of X given the set of measurements Z, which can be reformulated by employing Bayes' theorem:

$$\underset{X}{\operatorname{argmax}}\, p(X|Z) = \underset{X}{\operatorname{argmax}}\, p(Z|X)p(X)$$

In case no prior knowledge for X is available, $p(X)$ can be interpreted as an uniform distribution and can therefore be dropped from the optimization except for the prior factor p_0, so that MAP reduces to maximum

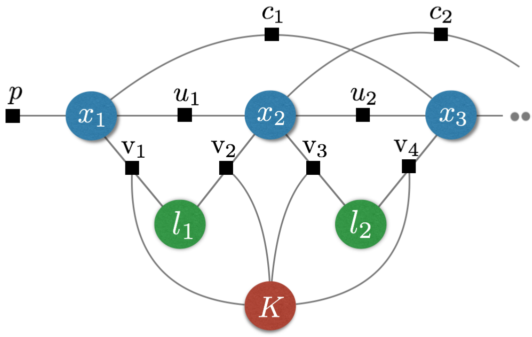


Figure 1: **SLAM represented as a factor graph.** Robot poses at consecutive time steps $\{x_i|i \in \mathbb{N}\}$ are drawn as blue circles, whereas sample points serving as landmarks $\{l_i|i \in \mathbb{N}\}$ – as green circles. The intrinsic camera parameters summarized in the camera matrix K are represented by the red circle. Each edge incident to a circle leads to a black square standing for a factor in the optimization. Odometry constraints are modeled by factors labeled with $\{u_i|i \in \mathbb{N}\}$, point observations – by factors labeled with $\{v_i|i \in \mathbb{N}\}$ and loop closures – by factors labeled with $\{c_i|i \in \mathbb{N}\}$. The prior factor p anchoring the first frame pose at the origin of the coordinate system ensures that the optimization problem is well-posed. Taken from [5].

likelihood estimation for the set of measurements. Under the assumption of conditional independence, the objective function can be reformulated as:

$$\underset{X}{\operatorname{argmax}}\, p_0\, p(Z|X) = \underset{X}{\operatorname{argmax}}\, p_0 \prod_{i=1}^{n} p(z_i|X) = \underset{X}{\operatorname{argmax}}\, p_0 \prod_{i=1}^{n} p(z_i|X_i)$$

where the measurement set has cardinality $|Z| = n$ and each measurement $z_i \in Z$ depends solely on the subset $X_i \subseteq X$ of optimization variables connected to its corresponding optimization factor. Assuming zero-mean Gaussian measurement noise with information matrix Ω_i for ε_i, the likelihood of a single measurement can be understood as:

$$p(z_i|X_i) \propto exp(-\frac{1}{2}||h_i(X_i) - z_i||^2_{\Omega_i})$$

where we measure length $|| \, ||_{\Omega_i}$ with respect to the metric induced by Ω_i. The prior factor can be formulated likewise as $p_0 \propto exp(-\frac{1}{2}||h_0(X_0) - z_0||^2_{\Omega_0})$ and the maximization of the posterior probability can be turned into a minimization of the negative logarithm thereof:

$$\underset{X}{\operatorname{argmin}} -ln\ (p_0 \prod_{i=1}^{n} p(z_i|X_i)) = \underset{X}{\operatorname{argmin}} \sum_{i=1}^{n} ||h_i(X_i) - z_i||^2_{\Omega_i}$$

which yields a non-linear non-convex least squares optimization problem. One of the main differences from the bundle adjustment formulation stated in a pure model of projective geometry as in Equation 5 is the generality of the SLAM framework, which enables the simultaneous incorporation of various sensors whose measurements are all represented as factors in the factor graph and considered jointly during optimization. SLAM can thus perform optimization for a broad variety of sensor models and their combinations employing measurements from visual, inertial, sonar, laser sensors, wheel encoders etc. at once. It is also distinct from bundle adjustment in that it can be solved incrementally as soon as new measurements arrive, which is essential for real-time applications such as the Mars exploration rover, autonomous drones or self-driving cars.

2.2.2 Classification of SLAM Frameworks

The classification of SLAM frameworks can be done along two main axes: direct vs. indirect and dense vs. sparse approaches. In the following, we describe these distinctions as done in [9]. Indirect methods can

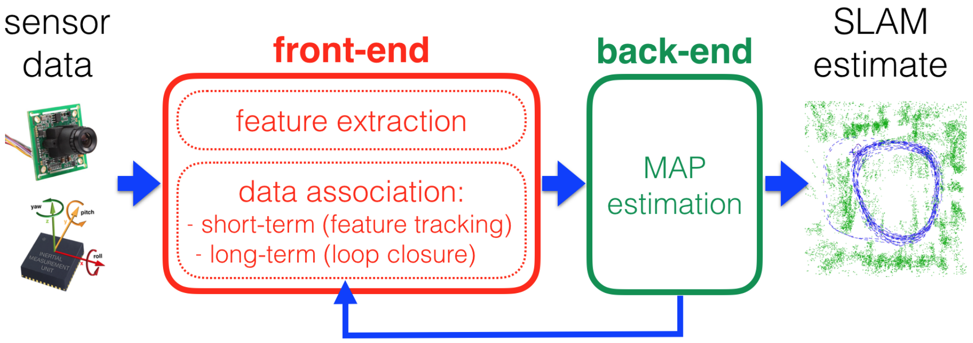


Figure 2: **Overview of a feature-based SLAM method.** The sensor data serves as an input for the front-end comprising feature extraction, descriptor computation and data association, on the basis of which the back-end can perform a MAP estimation for the camera poses at given keyframes and the positions of observed points in three-dimensional space. The back-end also invokes the front-end so as to obtain data associations used for loop detection. Taken from [5].

be conceptually separated into a front-end and a back-end as shown in Figure 2. In the case of a feature-based method, the front-end first extracts easily distinguishable features such as corners, e.g. by means of SIFT [25]. Their descriptors enable reliable matching in the presence of arbitrary rotations, scale and photometric variations, which facilitates data association for points observed in several different keyframes. The back-end then performs MAP estimation as explained in Subsubsection 2.2.1.

In a direct method, the front-end is left out altogether. Non-visual sensor data can still be used for the optimization of a geometric error function, whereas a photometric error modeling the full image formation process is defined for passive visual sensors. In the case of a simple camera, the photometric error formulation makes use of the intensities measured at sample points with high image gradients. More intricate visual sensors can provide richer information such as the irradiance and the exposure time for each pixel, as well as the lens attenuation and the response function, non-linear in case of gamma correction or white balancing, which are all incorporated into the error definition.

Indirect methods have been prevalent in the last decade due to their efficiency and precision. On the contrary, direct methods are only recently gaining popularity due to their simpler formulation and their ability to use more ubiquitous image information, allowing pixel-wise depth estimation in low-textured environments.

Sparse methods only consider a set of environmental landmarks such as corners, whose positions are assumed to be uncorrelated. This induces the absence of a geometry prior resulting in the lack of a neighbourhood relation and depth gradient smoothness, which can in turn be achieved by dint of dense methods. They are also able to reconstruct a dense map of the explored environment.

2.3 Deep Learning

Single-layer neural networks are equivalent to generalized linear discriminants. Their task is to learn optimal weights for the connections between the input and the output layer e.g. so as to be able to separate instances of different classes. Even though they can employ non-linear activation functions, these are fixed and cannot be learned, resulting in a cumbersome feature design for each specific application scenario. With the introduction of layers of hidden units, whose connection weights are also adjusted, feature design becomes part of the learning process. While any continuous function over a compact domain can theoretically be approximated arbitrarily well with a single hidden layer as shown in [21], it relies on the availability of an infinite number of hidden units. Deep neuron networks have been developed as an alternative relevant for the solution of practical tasks.

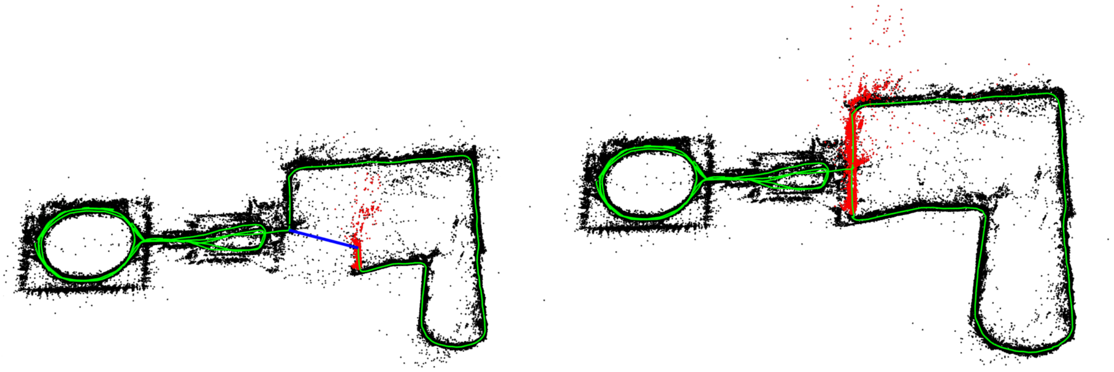
A hierarchical multi-layered neural network structure has been tailored to the needs of the computer vision domain thus introducing convolutional neural networks dramatically reducing the number of learned parameters. Further improvements included the alternation of convolutional and subsampling layers, as well as the inclusion of one or several fully connected layers preceding classification and deconvolutional layers for image segmentation. The classification loss can be defined in several ways, e.g. by means of softmax activation with a cross-entropy loss. Rectified Linear Unit (ReLU) activations as the simplest form of a non-linear activation function are used for the hidden layers with the aim of modeling biological neurons more realistically by only allowing non-negative values. More importantly, they do not lead to vanishing gradients during weight backpropagation.

GoogLeNet presented in [40] has broken up with the prevalent network architecture of sequentially stacked layers by inventing the inception module as a novel building block enabling improved performance and more efficient network training. Residual networks with skip connections have rendered even deeper networks possible, whose training is still feasible, while yielding higher precision as outlined in [19]. This stems from their effective behavior as an ensemble of shallow networks as expounded in [41]. Convolutional neural networks have also been successfully employed for object detection [33], semantic segmentation [37] etc.

In the context of SfM, deep convolutional learning networks for monocular depth, see Subsection 3.3 and [14,23,45], camera motion, see Subsection 3.4, optical flow estimation, see [46] and rigid body motion, see SE3-Nets [4], have enabled the construction of methods that optimize a mixed geometric and photometric loss function purely by learning. Under the assumption of a rigid environment and often interpreting dynamical object motion merely as a noise source, they aim at a dense association of pixels from consecutive frames in order to estimate the pixel depths. In conjunction with ego-motion tracking, an optical flow estimate can also be delivered. It is more precise in the presence of rigid body motion estimates for dynamical objects observed in the scene. The loss function formulations are analogous to the ones employed by direct SLAM frameworks, albeit their optimization is performed over neural network weights and not over common projective geometry model parameters such as intrinsic and extrinsic camera parameters or optical flow vectors. They are therefore able to improve as long as they are being trained on various video input and thus become resilient to sparsely textured regions and degenerate camera motion.

The advanced concepts of keyframes, local optimization windows and loop closure are not employed in neural networks yet. Instead, they typically estimate depth from a single frame, as well as camera and rigid body motion from a pair of consecutive video frames. The motion estimation is done under the assumption of a rigid scene, rarely taking rigid body motion into account. For a notable exception, cf. Subsection 3.5. The pinhole camera model and its corresponding image formation process, as well as the laws of projective geometry are used to constrain the dimensionality of the SfM solution space to six degrees of freedom for rigid body motion and one degree of freedom for per-pixel depth. The inversion of the image formation process is an ill-posed problem since a multitude of spatial configurations lead to the same pixel irradiance. This issue can be handled either by a partial supervision of network connection weights or by pre-training on a wide variety of visual input. Since the same network weights account for the consistent perception of a multitude of video input, the network should be able to circumvent the ambiguities of the SfM problem. This has been confirmed empirically by the successful development of frame-to-frame motion field estimation under supervised learning [11], semantic label propagation [32] and temporally coherent visual feature extraction [43,44] suitable for object detection and classification under unsupervised learning, just to name a few examples.

Deep learning builds upon the concepts of non-linear optimization since the training of a multi-layer network constitutes a high-dimensional, non-convex, non-linear optimization problem. In the presence of predominantly saddle-shaped regions and a multitude of local optima, training can take a lot of time and there is no transparency as to whether the optimum has already been reached. Batch normalization is one known remedy that reduces the duration and improves the stability of network training by shifting the mean output activation of the whole layer to zero. It is also prone to overfitting, such that the training set error decreases further at the cost of an increasing test set error, thus degrading the generalization abilities of the neural network. Dropout and the use of a validation set during network training have been recognized as feasible solutions. For a more in-depth treatment of general machine learning concepts and deep learning in particular, the reader is kindly referred to [3] and [17], respectively.



(a) Loop detection

(b) Loop closure

Figure 3: **Loop detection and closure in ORB-SLAM.** The estimated trajectories before and after loop closure are shown in green, whereas red points represent three-dimensional point positions estimated over the course of the loop. On the left-hand side, both ends of the detected loop are connected by a blue edge. On the left-hand side, they have already been aligned to each other, which has also lead to a shift of the red points to their globally consistent positions. Taken from [28].

3 Related Work

In this section, we present a multitude of approaches developed in recent works for a wide spectrum of SfM formulations. They should give the reader an overview of the recent research in the field and thus provide the means for a correct placement of the discussed method among its competitors.

3.1 ORB-SLAM2

A feature-based sparse monocular SLAM method has been proposed in [28] and revised in [30]. It maintains a covisibility graph, in which keyframes with a substantial number of coinciding point observations are connected by an edge, and consists of three concurrently executed threads. The tracking thread is responsible for tracking the camera motion. The pose at the current frame is estimated by means of motion-only bundle adjustment with fixed three-dimensional point positions. It is initialized as the SE(3) transformation obtained by feature matching against the previous keyframe. The estimated pose of the new frame is forwarded to the depth map estimation thread, on the basis of which it decides whether or not to turn it into a keyframe so that it would later on be used as a tracking reference in the tracking thread. In case it does not become a keyframe, it is used for pose refinement of the currently processed keyframe and all other keyframes adjacent to it in the covisibility graph, as well as for a position update of observed points. This is done by means of local bundle adjustment.

New points are triangulated on the basis of ORB descriptor matches between pairs of adjacent keyframes in the covisibility graph. In order to maintain the optimization sparsity and thus accelerate the local bundle adjustment, redundant keyframes and points observed by few keyframes are purged from the optimization. In case of lost tracking, global relocalization is performed.

The map optimization thread performs loop detection and closure as outlined in [29]. In each keyframe, the scale- and rotation-invariant ORB feature descriptor presented in [35] is computed for the interest points delivered by FAST corner extraction, see [34], forming the bag-of-words vector of the keyframe. It is inserted into a DBoW2 bag-of-words keyframe database, see [13], and matched against the bag-of-words vectors of previous keyframes so as to recognize loop detection candidates. A three-dimensional similarity transformation estimation interleaved with RANSAC [12] serves as loop verification. In case of a valid loop detection, a relevant subset of the covisibility graph is optimized for global consistency. This counteracts the accumulated motion estimation drift, resulting in the alignment of both loop ends, the correction of triangulated point positions over the course of the loop and the fusion of duplicated points as visible in Figure 3.

3.2 DSO

The direct dense monocular odometry framework presented in [9] outperforms the tracking thread of ORB-SLAM2 in terms of the ego-motion estimation accuracy given photometrically well-calibrated data such as the dataset introduced in [10]. It first photometrically corrects each video frame by inferring the irradiance $I_i[\mathbf{p}]$ at each pixel $\mathbf{p}$ of frame i from the given lens attenuation and non-linear response function. It then uses the irradiance at pixels from regions with large intensity gradients such as corners, edges and smooth intensity variations on walls to define a photometric error, which also relies on the exposure time t_i of each frame. In the case of its absence, it is modeled by a brightness transfer function $bt : I_i \mapsto e^{-a_i}(I_i - b_i)$, which estimates the optimal parameters a_i and b_i for each frame i with irradiance I_i. The photometric error of a single point reprojection from keyframe i into keyframe j is then defined as follows:

$$E_{\mathbf{p}ij} = \sum_{\mathbf{q} \in N_{\mathbf{p}}} \|(I_j[\mathbf{q}'] - b_j) - \frac{t_j e^{a_j}}{t_i e^{a_i}}(I_i[\mathbf{q}] - b_i)\|_\gamma \tag{6}$$

where $\mathbf{p}$ is the point to be projected, $N_{\mathbf{p}}$ – a residual pattern of neighbouring points of $\mathbf{p}$ and $\mathbf{q}'$ is defined as in Equation 4. $\|\|_\gamma$ stands for the Huber norm.

A local optimization window w of active keyframes is continuously maintained so as to reflect changes in the field of view, occlusions and disocclusions, as well as considerable changes in the exposure time. Its photometric error is formulated as:

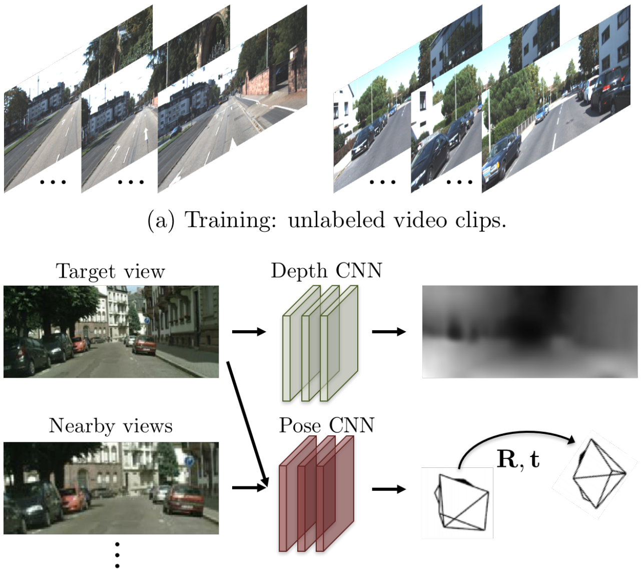
$$E_w = \sum_{i \in F} \sum_{\mathbf{p} \in P_i} \sum_{j \in obs(\mathbf{p},i)} E_{\mathbf{p}ij} \tag{7}$$

where F denotes the set of active keyframes, P_i – the set of points in keyframe i and $obs(\mathbf{p},i) \subseteq F \setminus \{i\}$ – the set of active keyframes other than i, into which each point from the residual pattern of $\mathbf{p}$ can be projected. This formulation enables the joint optimization of the camera intrinsic and extrinsic parameters, as well as the photometric ones used for brightness transfer. The three-dimensional points are bound to the frame in which they have been observed and their depths are also optimized in a coarse-to-fine manner. At each of the smaller resolutions, pixel intensities are interpolated bilinearly. The photometric error is then minimized in an iterative manner by means of linearization at the current estimate. The Levenberg-Marquardt algorithm is employed for enhanced stability.

The optimization efficiency, essential for real-time operation, depends greatly on the sparsity of the linear systems at each iterative step. This is the reason why sampled points not having been observed in the most recent keyframes are continuously marginalized until too few of them remain active. At this point, their holding keyframe is recognized as having a pose too different from the ones of the most recent keyframes. Its pose is thus marginalized and it is removed from the local optimization window. Neither loop closure nor duplicate point fusion is performed. The optimization is done exclusively locally and hence on the basis of SE(3) transformations not accommodating scale drift correction. Nevertheless, it can deal with various indoor and outdoor environments inducing fluctuating exposure times and big scale changes over the course of the trajectory and still accumulate a relatively small motion estimation drift.

3.3 Unsupervised Monocular Depth Estimation

An important unsupervised monocular depth estimation framework has been introduced by Godard and al. in [16]. Their main contribution is the introduction of a loss term enforcing left-right consistencies for monocular depth estimation trained on stereo image pairs. Epipolar constraints and a known stereo baseline are employed for the generation of disparity images in both directions for any given stereo image pair, thus transforming the depth estimation into an image reconstruction problem. The loss function also includes terms encouraging structural similarity of warped image patches and depth disparity smoothness. Network training and evaluation are performed on non-labelled stereo datasets such as KITTI. Training on the Cityscapes dataset and fine-tuning the network parameters on the dataset under evaluation in a self-supervisory manner has been recognized as a potential depth prediction accuracy booster. The proposed unsupervised deep network has shown an improved performance in comparison with state-of-the-art supervised learning approaches. However, the method is not able to deal with occlusions, cannot be trained on



(a) Training: unlabeled video clips.

(b) Testing: single-view depth and multi-view pose estimation.

Figure 4: **SfMLearner overview.** Based on unlabeled video input, the SfMLearner delivers a per-pixel depth estimate for each single frame, as well as a relative pose estimate for a pair of consecutive frames. This is done by means of two independent convolutional neural networks as shown in greater detail in Figure 5. Taken from [47].

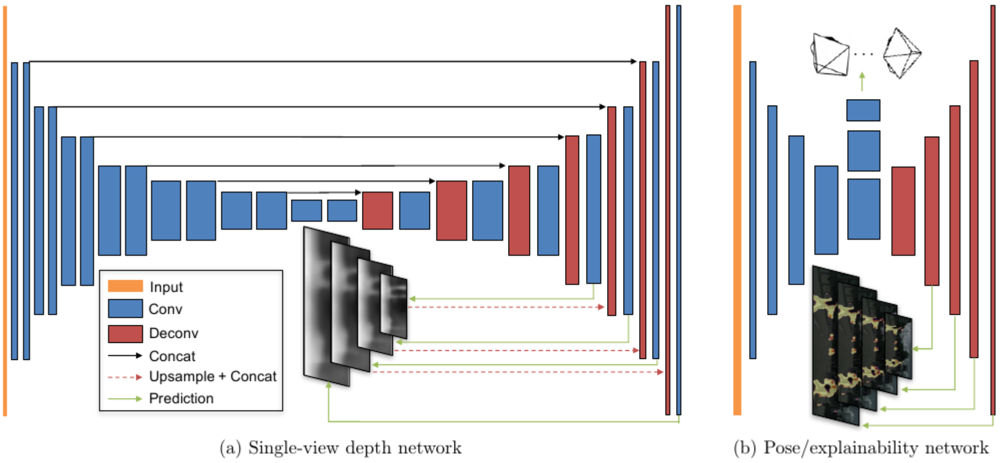
monocular video and does not consider temporal consistency between frames so that it effectively regards a pair of images taken from a stereo rig independently of the video sequences they belong to.

3.4 Unsupervised Monocular Depth and Ego-Motion Estimation

An approach tackling unsupervised learning of depth and ego-motion from monocular video has already been proposed by Zhou et al. in [47]. It has also been one of the first research papers presenting unsupervised monocular depth estimation. An overview of the method is depicted in Figure 4. Warping adjacent frames towards each other based on estimated pixel depths and camera ego-motion has enabled the optimization of a differentiable photometric loss function. A further loss term is used for the generation of a per-pixel soft mask suppressing pixel locations, at which the optical flow cannot be explained by ego-motion alone e.g. because of moving objects in the observed scene, occlussions and disocclusions etc. A least absolute deviations depth gradient smoothness term has also been included in the objective function, each of whose terms is evaluated at multiple scales. The network architecture is shown in Figure 5. The structure subnetwork used for depth estimation is adopted from DispNet, see [27]. Even though this approach does not benefit from any supervision, it has been proven to perform competitively in comparison with supervised deep networks and not to exceed by far the best evaluation metrics to-date obtained with full SLAM systems such as ORB-SLAM.

3.5 SfM-Net

SfM-Net is a SfM deep learning network proposed in [42]. It builds upon previous work on the mostly supervised learning of monocular depth and the optical flow field and, most notably, employs differentiable image warp inspired by the differentiable spatial transformer module proposed in [22]. It explicitly models the rigid body motion of a fixed number of objects by estimating an object mask and the object motion corresponding to it. In case fewer objects than this fixed number are present in the observed scene, the superfluous object masks simply remain empty, while a higher number of objects means some of them remain untracked and are thus reduced to noise sources. The object masks may overlap, thus enabling motion composition, which is e.g. useful for the modeling of kinematic chains that are able to describe the



(a) Single-view depth network

(b) Pose/explainability network

Figure 5: **SfMLearner architecture.** The structure subnetwork is depicted on the right-hand side, whereas the motion subnetwork additionally delivering a mask of pixels valid for reprojection is drawn on the left-hand side. The width of each rectangular block represents the output dimension, while its height indicates the spatial dimension of the feature map at a given network layer. Each apparent change in height and width corresponds to a reduction or increase of dimensionality by a factor of two. The structure subnetwork performs multi-scale single-view depth estimation. The motion subnetwork has two branches, the first of which does dual-view relative pose estimation. Its second branch is responsible for the production of multi-scale explainability masks used for pixel reprojection. Taken from [47].

motion of deformable objects. If no object masks applies at a given pixel position, it is interpreted as part of the background and assumed to stay fixed until the image at the next frame is taken, so that its perceived motion can be modeled as a function of the camera motion alone.

In the following, we discuss the network architecture, based on [27] and shown in Figure 6. There are two identical subnetworks, each of which is trained with different spatial smoothness priors and independent connection weights. In the motion subnetwork, an L1-norm penalty is imposed upon the gradients of the optical flow field, the motion masks and depth across adjacent pixels, whereas in the structure subnetwork, this penalty is imposed upon second-order depth gradients so as to enforce non-constant, spatially smooth depth value predictions. The motion subnetwork is forked at its embedding layer into a motion prediction and an object mask estimation branch. The motion prediction branch delivers separate estimates for the camera ego-motion and each rigid body motion, whereas the object mask estimation branch provides a segmentation into pixels belonging to the object and the rest of them.

The motion estimation process is represented in Figure 7. The depth at each pixel is predicted monocularly and used for a three-dimensional point cloud generation. In conjunction with the ego-motion and several rigid body motion estimates in each direction, this enables an independent uni-directional estimation of the optical flow field between two adjacent video frames. Differentiable warp based on the estimated optical flow field is used for the association of pixels whose depths must coincide. Inconsistencies between the depth predictions for corresponding pixels in consecutive frames are punished by means of a forward-backward loss inspired by the left-right consistency constraints described in Subsection 3.3 in order to enforce temporal consistency of the depth estimates. The motion subnetwork used for forward and backward image warp is the same and thus connection weights are shared. The estimated optical flow field is also used for photometric loss computation.

Self-supervised learning is made possible by explicit geometric and photometric loss formulations based on the estimated optical flow field. Rotations are parametrized by means of three Euler angles describing rotations around each coordinate system axis and a pivot point. Let P denote the set of pixels with cardinality $|P| = w \cdot h$, where the frame dimensions are described by width w and height h. Following the projective geometry notation conventions laid down in Subsection 2.1.1, the pixel $\mathbf{p}_t^i \in P$ can be projected back into three-dimensional space by means of:

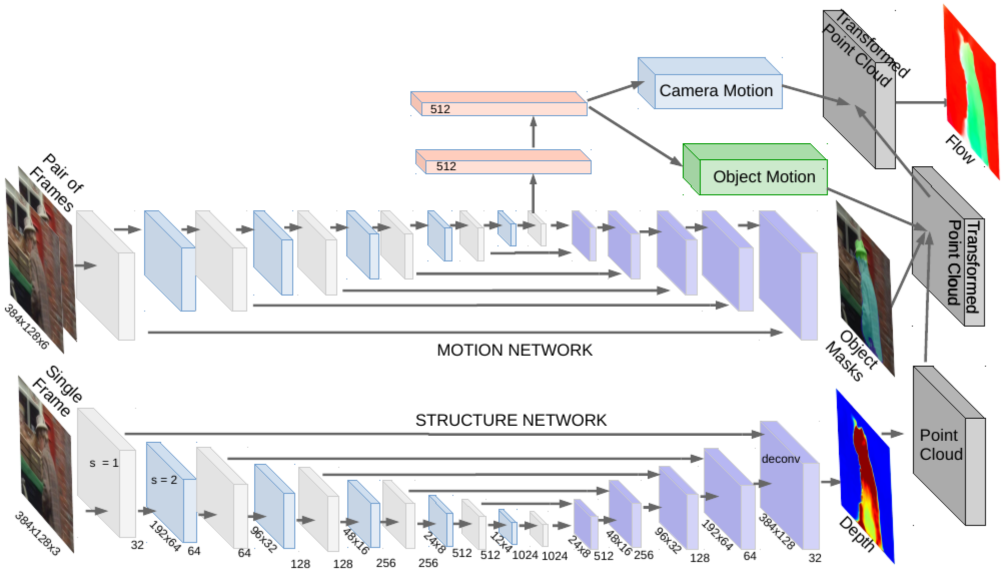


Figure 6: **SfM-Net architecture.** Each of both identical subnetworks consists of alternating 3 x 3 convolutional layers with stride one and two, respectively, at whose output batch normalization is applied. They are followed by several deconvolution layers serving for an upsampling back to the input frame size, each of which is concatenated with the feature maps of the convolutional layer with the same dimensions. Depth values are predicted by ReLU activations with a bias towards the value 1 and clipped at the value 100 so as to provide useful depth gradients. Object masks are predicted by sigmoid activations so as to allow simultaneous membership in several object masks and their logits are multiplied by a value proportional to the iteration step so that sharp masks are obtained. The pivot variables modeling the centers of rotation of the camera and each dynamic object are computed as a weighted average of the coordinates of all pixels activated by the softmax function. Two fully-connected layers drawn in pink precede the camera and dynamic object motion estimation in an additional network branch emerging from the motion subnetwork layer with the coarsest feature maps. Taken from [42].

$$p_{it} = K^{-1}\Pi^{-1}(\mathbf{p}_t^i, d_i(\mathbf{p}_t^i))$$

where $d_i : P \rightarrow [1, 100]$ maps a pixel to its depth estimate. First comes the application of each estimated dynamic object rotation R_t^k with pivot point p_t^k and translation t_t^k for each object k at time t with corresponding Boolean object mask function $m_t^k : P \rightarrow \{0, 1\}$:

$$p'_{it} = p_{it} + \sum_{k=1}^{K} m_t^k[\mathbf{p}_t^i]\ (R_t^k(p_{it} - p_t^k) + t_t^k - p_{it})$$

Then the estimated camera motion consisting of rotation R_t^c with pivot point p_t^c and translation t_t^c is applied:

$$p''_{it} = R_t^c(p'_{it} - p_t^c) + t_t^c$$

Projecting the transformed point into the frame results in:

$$\mathbf{p}_{t+1}^i = \Pi(Kp''_{it})$$

The motion vector estimate for pixel i is thus $\mathbf{p}_{t+1}^i - \mathbf{p}_t^i$. The photometric loss is then defined as:

$$E_{ph}^t = \frac{1}{w \cdot h} \sum_{\mathbf{p}_t^i \in P} |I_t[\mathbf{p}_t^i] - I_{t+1}[\mathbf{p}_{t+1}^i]| \tag{8}$$

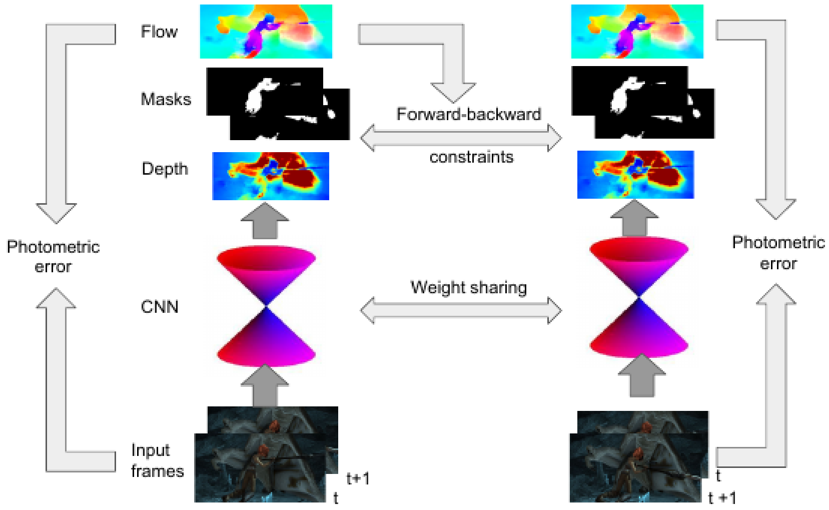


Figure 7: **Motion subnetwork overview.** The motion subnetwork operates on a pair of successive video frames. The per-pixel depth value predictions for each frame delivered by the structure subnetwork from Figure 6 are used to project each pixel back into three-dimensional space and thus obtain a dense point-cloud. It is transformed first according to each dynamic object motion estimate and subsequently according to the camera motion estimate so as to reflect the predicted scene flow in forward (left-hand side) and backward direction (right-hand side). The projection of the transformed point cloud into the other frame enables a two-dimensional optical flow field estimate, which is in turn used for a differentiable warp between both frames. Gradients for the photometric loss defined in Equation 8 are computed based on the differences in the original and the warped pixel intensities, whereas the depth smoothness gradients for the optimization of the forward-backward consistency loss from Equation 9 are obtained by enforcing temporal consistency between subsequent frames. Taken from [42].

with I_i designating the irradiance at frame i. The forward-backward loss used to enforce temporal depth consistency is formulated as:

$$E^t_{fb} = \frac{1}{w \cdot h} \sum_{i \in P} |d_i(\mathbf{p}^i_t) + \Delta Z_t(\mathbf{p}^i_t) - d_{i+1}(\mathbf{p}^i_{t+1})| \tag{9}$$

where $\Delta Z_t(\mathbf{p}^i_t)$ denotes the z-coordinate of the motion flow at pixel $\mathbf{p}^i_t$ so that $\Delta Z_t(\mathbf{p}^i_t) = [p''_{it} - p_{it}]_{3,1}$.

Depth and camera motion supervision can be added for the training and evaluation on the RGB-D SLAM dataset [38]. The depth loss term is defined as:

$$E^t_{depth} = \frac{1}{w \cdot h} \sum_{\mathbf{p}^i_t \in P} m^{gt}_t[\mathbf{p}^i_t] \; |d_i(\mathbf{p}^i_t) - d^{gt}_i(\mathbf{p}^i_t)|$$

with ground-truth depth value $d^{gt}_i(\mathbf{p}^i_t)$ at pixel $\mathbf{p}^i_t$ from frame i at time t available only at the positions marked as true by the Boolean mask functions $m^{gt}_t : P \to \{0,1\}$.

In the presence of a ground-truth trajectory, the camera pose at time step t can be computed as a frame-to-frame rotation R^{c-gt}_t, followed by a translation t^{c-gt}_t. The relative transformations with respect to the estimated rotation R^c_t and translation t^c_t can be computed as:

$$R^{err}_t = (R^c_t)^T \; R^{c-gt}_t$$

$$t^{err}_t = (R^c_t)^T \; (t^c_t - t^{c-gt}_t))$$

The corresponding loss functions are defined based on [38] as:

$$E^t_{c_{rot}} = arccos(min(1, max(-1, \frac{1}{2}(trace(R^{err}_t) - 1))))$$

$$E^t_{c_{tr}} = \|t^{err}_t\|_2$$

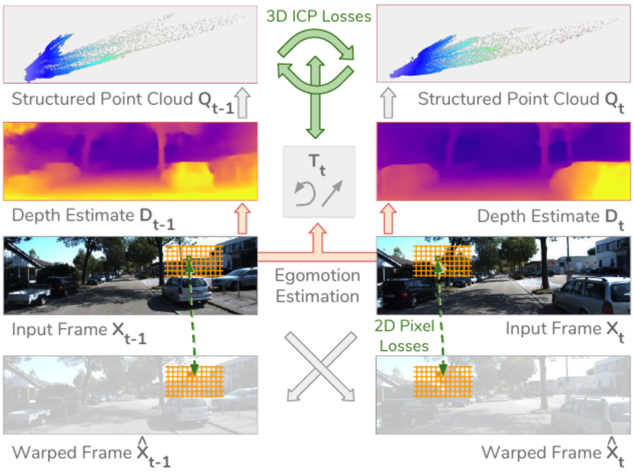


Figure 8: **Method summary.** The depth images are samples estimated by the trained network. Together with the camera motion estimate, they are connected to the input by orange arrows representing prediction steps. Spatial transformation steps relating an input frame to an alternate version of its warped in time are depicted as gray arrows. The computation steps of the three-dimensional point cloud alignment loss, as well as the two-dimensional photometric, structured similarity and depth smoothness losses are drawn in green. Taken from [26].

4 Method

The method introduced in [26] employs a direct dense formulation for the unsupervised learning of monocular depth and ego-motion estimation. Its most notable novelty is the introduction of a differentiable three-dimensional geometry loss term enforcing consistent point cloud reconstructions across consecutive frames. It is combined with a dense two-dimensional photometric loss term based on optical flow estimation, a depth smoothness loss term and a structured similarity loss term so as to define a novel differentiable objective function suitable for weight backpropagation in deep neural networks.

4.1 Method Overview

The main steps of the approach are depicted in Figure 8. Taking two consecutive frames as an input, the deep network provides a depth estimate for each of them independently in a monocular fashion and an ego-motion estimate by taking into account both frames at once so as to estimate the optical flow between them. Each frame is warped towards the other one in order to compute the photometric and the structured similarity loss. The point clouds obtained by projecting pixels with estimated depths into three-dimensional space are aligned by the ego-motion estimate and their spatial inconsistencies are penalized by the three-dimensional geometry loss term.

In the optical flow field estimation, some pixels may be mapped to positions outside of the frame boundaries. This issue is even more paramount with forward ego-motion and a backward warp. Therefore masks of pixels valid for reprojection have to be computed so as to handle parallax effects and objects entering or leaving the field of view. It is important to prevent reprojections at unreliable pixel positions that would otherwise degrade the optimization. Since validity mask learning is not very effective in practice, in contrast to [47], the proposed method analytically computes the Boolean masks of valid pixels $M_i : P \to \{0,1\}$ in frame i and thus slightly simplifies the learning problem.

4.2 Loss Function

The loss function is comprised of the following four terms, the first three of which are two-dimensional: a photometric consistency term, a structured similarity term, a depth gradient smoothness term and a three-dimensional point cloud alignment term. The photometric consistency term is derived by pixel reprojection in a direct dense fashion similar to the one employed in Equation 8:

$$L_{ph} = \sum_{\mathbf{p}_t^i \in P} \|M_i[\mathbf{p}_t^i](I_t[\mathbf{p}_t^i] - I_{t+1}[\mathbf{p}_{t+1}^i])\|_2$$

Since the pixel position $\mathbf{p}_{t+1}^i$ is not an exact one, a bi-linear interpolation of the neighbouring four pixel intensities is performed. Note that the pinhole camera and the prospective projection model are designed preeminently for differentiability in mind and unfortunately turn out to be crude approximations of the image formation process neglecting e.g. shadows, translucence and reflections. The photometric loss is therefore noisy and a strong regularization would only deliver smoothed predictions. A direct adjacent frame prediction would circumvent this issue, but would not provide depth and ego-motion estimates essential for SfM.

The structured similarity term also used in [16] is based on image patches from two consecutive frames. Structured similarity for image patches centered at pixels x and y with local means and variances μ_x, σ_x and μ_y, σ_y and covariance σ_{xy} computed by simple fixed pooling or convolution is defined as:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x + \sigma_y + c_2)}$$

where c_1 and c_2 are small constants. The corresponding loss term for minimization is defined as:

$$L_{SSIM} = \sum_{\mathbf{p}_t^i \in P} M_i[\mathbf{p}_t^i](1 - SSIM(I_t[\mathbf{p}_t^i], I_{t+1}[\mathbf{p}_{t+1}^i]))$$

The depth gradient smoothness term used in [16] as well ensures the preservation of the spatial continuity of rigid objects in space. It can be understood as a refinement of the depth smoothness loss employed in [47] and is formulated as:

$$L_{sm} = \sum_{\mathbf{p}_t^i \in P} \|\partial_x d_i(\mathbf{p}_t^i)\| e^{-\|\partial_x I_t[\mathbf{p}_t^i]\|} + \|\partial_y d_i(\mathbf{p}_t^i)\| e^{-\|\partial_y I_t[\mathbf{p}_t^i]\|}$$

where the L2 norm is used throughout, whereas ∂_x and ∂_y stand for the partial derivatives along the x and the y coordinate axis, respectively.

Derived by iterative closest point matching, the three-dimensional point cloud alignment term represents the main contribution of the paper under discussion. It enables the backpropagation of three-dimensional spatial inconsistency information invaluable for the refinement of ego-motion and depth estimates. Two three-dimensional point clouds obtained from two consecutive frames are iteratively aligned in order to minimize the 3D loss term and thus provide an SE(3) transformation that can be used for ego-motion estimation refinement, while its residual serves for a further improvement of the predicted pixel depths.

As shown in Figure 9, the pixels at two consecutive frames are independently projected into three-dimensional space with reference to their estimated depths. Two point clouds:

$$Q_t^i = \{M_i[\mathbf{p}_t^i](K^{-1}\Pi^{-1}(\mathbf{p}_t^i, d_i(\mathbf{p}_t^i)))|\mathbf{p}_t^i \in P\}$$

and Q_t^{i+1} are thus obtained, each of which is warped towards the other one with respect to the estimated forward ego-motion T_t so as to obtain the warped point clouds $Q_{t-1}^{i+1} = T_t Q_t^i$ and $Q_{t+1}^i = (T_t)^{-1} Q_t^{i+1}$. Point correspondences are established by means of nearest neighbour search and point-to-point distances between corresponding points in both point cloud pairs Q_{t-1}^{i+1} and Q_t^{i+1}, as well as Q_{t+1}^i and Q_t^i, are minimized by means of the Iterative Closest Point (ICP) algorithm [2, 6, 36]. Let $c : P \rightarrow P$ be the point-to-point correspondance function. The transformation T' that best aligns both point clouds is found by minimization of the following objective function:

$$\underset{T'}{\text{argmin}} \frac{1}{2} \|T' Q_{t-1}^{i+1} - Q_t^{i+1}\|_2^2$$

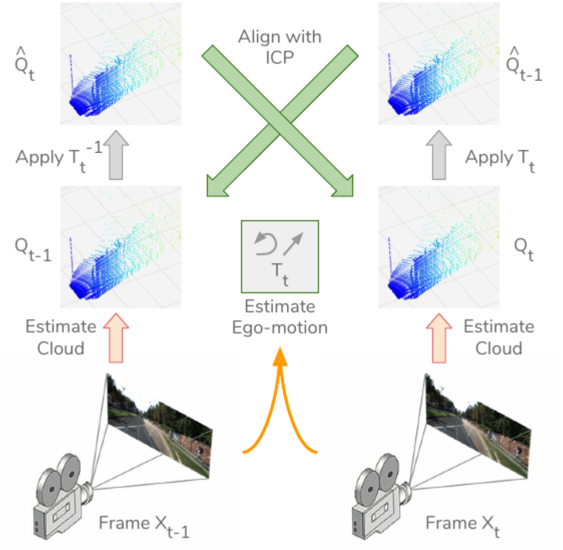


Figure 9: **Three-Dimensional Point Cloud Alignment Term.** This picture should be regarded as a more detailed view of the three-dimensional geometry loss term computation depicted as a single step in Figure 8. The point cloud at each frame is estimated from the predicted depth values at pixel positions validated by the reprojection mask. By warping the point cloud of each frame in the pair towards the other one by means of the ego-motion estimate, two point clouds can be found in the local coordinate system of each frame. They are brought to spatial coherence as the three-dimensional geometry loss term is minimized. This in turn provides approximate gradients for the ego-motion and depth estimation refinement. Taken from [26].

for the forward warp shown in Figure 10. Note that Q_{t-1}^{i+1} corresponds to $\hat{Q}_{t-1}$. The residual r_t at pixel $\mathbf{p}_t^i$ is then $T' Q_{t-1}^{i+1} - Q_t^{i+1}$. The three-dimensional geometry loss is defined as:

$$L_{3D} = |T_t' - I| + |r_t|$$

where $I \in \mathbb{R}^{4x4}$ denotes the identity matrix. Since T' points in the direction of the minimum found by ICP, it is used to adjust T_t and $d_i(\mathbf{p}_t^i)\ \forall \mathbf{p}_t^i \in P$. r_t is used as a futher approximation to the negative loss gradient with respect to the depth map. The ego-motion estimate is thus augmented by the transformation between both point clouds obtained by ICP and is subsequently adjusted together with the depth of each point from the source frame during the three-dimensional geometry loss term minimization so as to minimize the residual distance to its corresponding point from the target frame. The point cloud alignment, as well as the propagation of approximate gradients for the refinement of the camera motion estimate between two consecutive frames and the per-pixel depth estimate for each frame is visualized in detail in Figure 10.

The loss function is defined as a weighted sum of all four loss terms evaluated at four different scales s, each of which halves the frame dimensions:

$$L = \sum_s \alpha L_{ph}^s + \beta L_{3D}^s + \gamma L_{sm}^s + \omega L_{SSIM}^s$$

The hyper-parameters α, β, γ and ω are not learned, but remain fixed during training. They are typically optimized by self-supervision on a validation subset of the dataset.

4.3 Network Architecture

The discussed method employs the SfMLearner architecture developed by Zhou et al. It consists of a fully convolutional camera motion estimation subnetwork operating on pairs of consecutive frames and a fully convolutional dense depth estimation subnetwork operating on single frames. It is trained by means of self-supervision in the TensorFlow [1] framework with an Adam optimizer. After each training epoch,

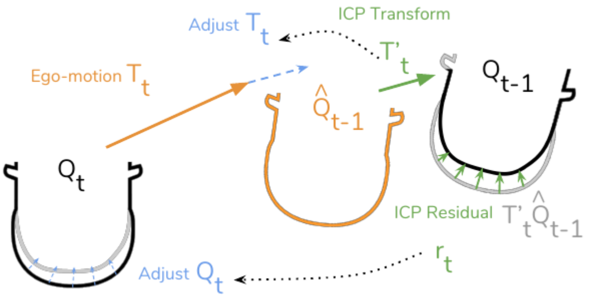


Figure 10: **Iterative point cloud alignment.** A bird's eye view over the front of a car. The orange arrow represents the point cloud warp based on the ego-motion estimate between both frames T_t. The ICP transform delivers a best-fit transformation update T_t' and a residual for each matched point pair, both drawn in green. The approximate gradients used for the update of the camera ego-motion and the three-dimensional position of each point in the warped point cloud are drawn in blue. Taken from [26].

the weights of each subnetwork are saved. The ones performing best on the validation set are later used for evaluation on the test set.

5 Evaluation

Evaluation is performed on the raw KITTI [15] and the Cityscapes [7] dataset, as well as on a custom dataset recorded with a hand-held mobile phone camera while driving a bicycle. The split of KITTI into a training, a validation and a test subset replicates the one already used in [47]. The depth estimation evaluation metrics include the Root Mean Square Error (RMSE) and the log RMSE error metric introduced in [8]. The Absolute Trajectory Error (ATE) defined in [39] is used for ego-motion estimation evaluation.

Following the notation from Subsection 3.5, the RMSE over the whole trajectory is defined as:

$$RMSE = \sqrt{\frac{1}{|P|}\sum_{\mathbf{p}_t^i \in P} \|d_i(\mathbf{p}_t^i) - d_i^{gt}(\mathbf{p}_t^i)\|_2^2}$$

In monocular depth prediction, the global scale of the observed environment is inherently ambiguous, however, the RMSE punishes deviations of the average scene scale from the ground truth. The scale-invariant log RMSE error has therefore been formulated as an alternative evaluation metric independent of the absolute global scale. Let $\Delta_i = ln\, d_i(\mathbf{p}_t^i) - ln\, d_i^{gt}(\mathbf{p}_t^i)$, then we can define:

$$RMSE^{log}_{scale-invariant} = \sqrt{\frac{1}{|P|}\sum_{\mathbf{p}_t^i \in P} \Delta_i^2 - \frac{1}{|P|^2}(\sum_{\mathbf{p}_t^i \in P} \Delta_i)^2}$$

An extensive comparison against the approaches presented in Subsection 3.4 and Subsection 3.5, as well as against various state-of-the-art supervised learning approaches can be seen in Table 1. The proposed method manages to outperform them all along a multitude of evaluation metrics and especially along the reliable scale-invariant log RMSE error.

The three-dimensional geometry loss ablation evaluation has shown the invaluableness of the novel three-dimensional geometry loss term for monocular depth and ego-motion learning, as can be seen in Figure 11. The ICP loss term acts as a regularizer thus mitigating overfitting. Qualitative results have also shown that it is able to reduce artifacts in sparsely textured image regions. Moreover, it considerably improves the efficiency of network training and the depth estimation quality.

Another more insightful training scheme consists of pre-training the network on a smaller dataset providing dense semantic object segmentation annotations such as Cityscapes and then fine-tuning the network connection weights on a multitude of unannotated videos such as the ones comprising KITTI in a self-supervised manner. This has indeed improved evaluation metrics as can be seen in Table 1.

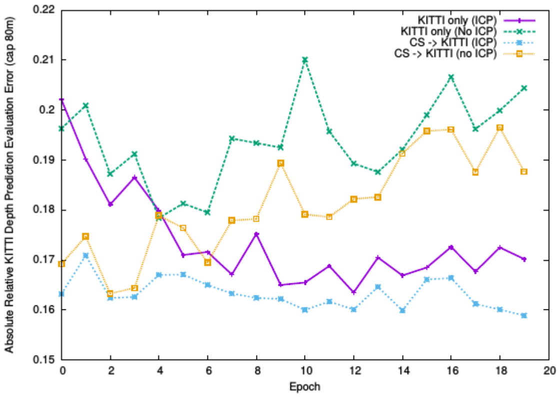


Figure 11: **Superimposed plots of depth error values after each of the training epochs.** It is clearly visible that the ICP loss term keeps the network from overfitting and thus improves its performance. Even a pre-training on the Cityscapes dataset could not compete with the inclusion of a three-dimensional term in the loss function. Taken from [26].

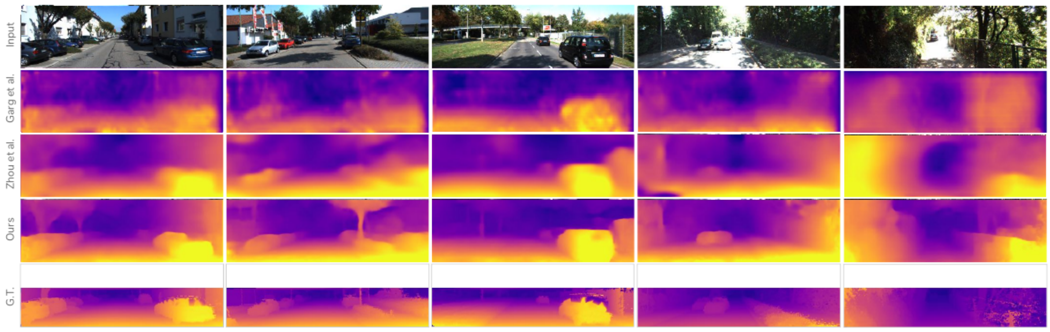


Figure 12: **Sample depth estimates delivered by several state-of-the-art monocular depth estimation approaches during evaluation on the KITTI dataset.** The depth maps produced by the approach lying at the heart of this report are closest to the ground truth annotations. Taken from [26].

Some qualitative results for depth estimation precision can be seen in Figure 12. Note that the proposed approach is able to estimate pixel depths correctly even in low-textured regions. It clearly outperforms both its unsupervised adversarial Zhou et al. and the supervised approach proposed by Garg et al. This can be explained by the imperfect alignment of ground-truth data and its corresponding camera image and the structural artifacts that LIDAR depth scanners used for the recording of the KITTI benchmark are known to produce in the presence of reflective, transparent or dark surfaces. The offset between the depth scanner and the camera may furthermore cause gaps in the point cloud projection into a camera frame. All of these issues have negative impact on the network training and thus also on the objective function minimization.

A truly outstanding achievement of the presented approach is its ability to learn from one dataset and still score competitive results when evaluated without any fine-tuning on another. This is all the more impressive if we consider the limitations of the bike dataset used for network training, recorded with a low-end hand-held mobile phone camera. The various differences between the custom bike dataset and the KITTI dataset such as the field of view discrepancy of 30 degrees, the lack of distortion correction vs. the full image rectification, wobbly vs. stable motion, the radically different architecture of the observed environment etc. have not prevented it from performing comparably to state-of-the-art methods as evident in Table 1. This result confirms the resilience of the proposed approach even under severe evaluation settings such as lens distortions and lack of stabilization.

In the following, an additional error metric for the camera motion estimation is presented. The pose estimate $P_i \in T$ at video frame i from the set of pose estimates T delivered by the tracking algorithm along

Table 1: $RMSE$ and $RMSE^{log}_{scale-invariant}$ of multiple depth estimation frameworks. Note that the results are better whenever the deep neural network is pre-trained on the Cityscapes dataset. The proposed method outperforms Zhou et al. even if the ICP term is taken out of the loss function, as well as in case it is only trained on KITTI, whereas Zhou et al. is pre-trained on Cityscapes and fine-tuned on KITTI. Note that it also outperforms various supervised methods such as Eigen et al. employing ground-truth depth supervision and Garg et al. employing stereo supervision during training. The SfM-Net evaluation from the paper did not provide any other evaluation metrics but the scale-invariant log RMSE. It has been evaluated on two different benchmarks from the KITTI dataset, while all other have been trained and evaluated on the raw KITTI dataset. It does not hold up to the proposed network trained on an unrelated bike dataset and evaluated directly without any fine-tuning on KITTI, which also manages to achieve results competitive to all other evaluated methods. Adapted from [26].

Method	Supervision	Dataset	Depth Cap	$RMSE$	$RMSE^{log}_{scale-invariant}$
All losses	–	Cityscapes + KITTI	0-80m	**5.912**	**0.243**
All losses	–	KITTI	0-80m	6.220	0.250
No ICP loss	–	KITTI	0-80m	6.267	0.252
Zhou et al.	–	Cityscapes + KITTI	0-80m	6.565	0.275
Zhou et al.	–	KITTI	0-80m	6.856	0.283
Eigen et al. Coarse	Depth	KITTI	0-80m	6.563	0.292
Eigen et al. Fine	Depth	KITTI	0-80m	6.307	0.282
All losses	–	Bike dataset	0-80m	7.741	0.309
No ICP loss	–	Bike dataset	0-80m	7.750	0.305
SfM-Net	–	Stereo KITTI 2012	0-80m	N/A	0.45
SfM-Net	–	Stereo KITTI 2015	0-80m	N/A	0.41
All losses	–	Cityscapes + KITTI	1-50m	**4.383**	**0.227**
All losses	–	KITTI	1-50m	4.549	0.231
Garg et al.	Stereo	KITTI	1-50m	5.104	0.273

Table 2: ATE on two sequences from the KITTI dataset. The proposed method clearly outperforms its competitor introduced by Zhou et al. and attains an equal or even a slightly better result than the full ORB-SLAM framework. Adapted from [26].

Method	Sequence 09	Sequence 10
Full ORB-SLAM	0.014 ± 0.008	**0.012 ± 0.011**
Zhou et al. [47]	0.021 ± 0.017	0.020 ± 0.015
No ICP loss	0.014 ± 0.010	0.013 ± 0.011
All losses	**0.013 ± 0.010**	**0.012 ± 0.011**

the course of the trajectory is modeled as a rigid-body transformation from camera to world coordinates. It first needs to be synchronized with its ground truth pose Q_i. The ATE error computation also needs an estimated transformation S mapping the tracked onto the ground-truth trajectory. Such a transformation can be obtained e.g. by means of the method presented in [20]. The ATE equals the RMSE of the translational components of the error transformations between the ground-truth pose and the warped pose estimate at each time step:

$$ATE = \sqrt{\frac{1}{|T|} \sum_{P_i \in T} \|trans(Q_i^{-1} S P_i)\|_2^2}$$

As visible in Table 2, the ego-motion estimate of the investigated approach significantly outperforms the one developed by Zhou et al. so that it is able to achieve results characteristic for SLAM systems that employ various supplementary mechanisms such as loop detection and closure, as well as global trajectory optimization to further improve their camera motion estimate. The tracking precision is once again improved by the inclusion of the three-dimensional geometry loss term.

6 Conclusion and Future Work

Depth and ego-motion estimation are essential for SfM and SLAM. They are irreplacable building blocks for motion planing and autonomous driving. While a carefully calibrated benchmark is required by most direct SLAM and supervised learning methods, which limits their application to a handful of established datasets such as KITTI and Cityscapes, unsupervised learning could potentially be done on any visual input input such as videos publicly available on the internet. Building solely upon simple assumptions such as temporal coherence in a mostly rigid environment and depth gradient smoothness along adjacent pixels, they are able to solve SfM on the basis of passive monocular vision. The proposed approach introduces a novel differentiable loss term that penalizes depth and ego-motion estimation inconsistencies in the three-dimensional space, thus optimally aligning three-dimensional point clouds in a common reference frame. This enabled it to outperform all relevant learning networks in the research field by even scoring equally to a full SLAM system in certain evaluation scenarios. It is also able to achieve competitive results even when trained on a highly irregular custom dataset captured by a widely available and inexpensive mobile phone camera and subsequently evaluated on a well-calibrated benchmark without any fine-tuning of the network parameters.

The integration of dynamic object detection and tracking would further improve the performance of the proposed approach as moving objects would no longer be considered as noise sources. The object motion and object mask estimation branches of the SfM-Net could therefore be introduced in the deep neural network. The validity masks for pixel reprojection can also be adjusted so as to consider occlusions and disocclusions of objects in the background brought about by viewpoint changes between adjacent frames. Furthermore, an optimization over an extended time lapse and not just for pairs of consecutive frames could enable higher-level visibility reasoning and therefore deliver robuster and more precise structure and motion estimates. Lastly, the scene model can be generalized so as to be able to also account for non-rigid objects and scale changes between successive video frames. By dispelling the Lambertian world assumption suitable exclusively for diffusely reflecting surfaces, the learning framework would also be able to handle specular reflections.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. A. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zhang. Tensorflow: A system for large-scale machine learning. *arXiv*, abs/1605.08695, 2016.

[2] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *ICRA*, 1992.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, 2006.

[4] A. Byravan and D. Fox. Se3-nets: Learning rigid body motion using deep neural networks. *arXiv*, abs/1606.02378, 2016.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE T-RO*, 2016.

[6] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *ICRA*, 1991.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. *CVPR*, 2016.

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *NIPS*, 2014.

[9] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *ICRA*, 2017.

[10] J. Engel, V. Usenko, and D. Cremers. A photometrically calibrated benchmark for monocular visual odometry. *arXiv*, abs/1607.02555, 2016.

[11] P. Fischer, A. Dosovitskiy, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. *arXiv*, abs/1504.06852, 2015.

[12] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *ACM*, 1981.

[13] D. Galvez-Lopez and J. D. Tardós. Bags of binary words for fast place recognition in image sequences. *IEEE T-RO*, 2012.

[14] R. Garg, V. K. B. G, and I. D. Reid. Unsupervised CNN for single view depth estimation: Geometry to the rescue. *arXiv*, abs/1603.04992, 2016.

[15] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 2013.

[16] C. Godard, O. Mac Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *arXiv*, abs/1609.03677, 2016.

[17] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[18] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv*, abs/1512.03385, 2015.

[20] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *OSA*, 1987.

[21] K. Hornik. Approximation capabilities of multilayer feedforward networks. *INNS*, 1991.

[22] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *arXiv*, abs/1506.02025, 2015.

[23] Y. Kuznietsov, J. Stückler, and B. Leibe. Semi-supervised deep learning for monocular depth map prediction. *arXiv*, abs/1702.02706, 2017.

[24] H. C. Longuet-Higgins. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*. Morgan Kaufmann Publishers Inc., 1987.

[25] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004.

[26] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. *CVPR*, 2018.

[27] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *arXiv*, abs/1512.02134, 2015.

[28] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós. ORB-SLAM: A versatile and accurate monocular SLAM System. *IEEE T-RO*, 2015.

[29] R. Mur-Artal and J. Tardós. Fast relocalisation and loop closing in keyframe-based SLAM. *ICRA*, 2014.

[30] R. Mur-Artal and J. D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *arXiv*, abs/1610.06475, 2016.

[31] O. Ozyesil, V. Voroninski, R. Basri, and A. Singer. A survey on structure from motion. *Acta Numerica*, 2017.

[32] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari. *Learning object class detectors from weakly annotated video*. 2012.

[33] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *arXiv*, abs/1506.01497, 2015.

[34] E. Rosten and T. Drummond. Machine learning for high-speed corner detection. *ECCV*, 2006.

[35] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. *ICCV*, 2011.

[36] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. *3DIM*, 2001.

[37] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE TPAMI*, 2016.

[38] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. *IROS*, 2012.

[39] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. *IROS*, 2012.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv*, abs/1409.4842, 2014.

[41] A. Veit, M. J. Wilber, and S. J. Belongie. Residual networks are exponential ensembles of relatively shallow networks. *arXiv*, abs/1605.06431, 2016.

[42] S. Vijayanarasimhan, S. Ricco, C. Schmid, R. Sukthankar, and K. Fragkiadaki. Sfm-net: Learning of structure and motion from video. *arXiv*, abs/1704.07804, 2017.

[43] X. Wang and A. Gupta. Unsupervised learning of visual representations using videos. *arXiv*, abs/1505.00687, 2015.

[44] L. Wiskott and T. J. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Comput.*, 2002.

[45] Z. Yang, P. Wang, W. Xu, L. Zhao, and R. Nevatia. Unsupervised learning of geometry with edge-aware depth-normal consistency. *arXiv*, abs/1711.03665, 2017.

[46] J. J. Yu, A. W. Harley, and K. G. Derpanis. Back to basics: Unsupervised learning of optical flow via brightness constancy and motion smoothness. *arXiv*, abs/1608.05842, 2016.

[47] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-motion from video. *CVPR*, 2017.