sensors

MDPI

Article

SVR-Net: A Sparse Voxelized Recurrent Network for Robust Monocular SLAM with Direct TSDF Mapping

Rongling Lang †, Ya Fan † and Qing Chang *

School of Electronics and Information Engineering, Beihang University, Beijing 100191, China
* Correspondence: changqing@buaa.edu.cn
† These authors contributed equally to this work.

Abstract: Simultaneous localization and mapping (SLAM) plays a fundamental role in downstream tasks including navigation and planning. However, monocular visual SLAM faces challenges in robust pose estimation and map construction. This study proposes a monocular SLAM system based on a sparse voxelized recurrent network, SVR-Net. It extracts voxel features from a pair of frames for correlation and recursively matches them to estimate pose and dense map. The sparse voxelized structure is designed to reduce memory occupation of voxel features. Meanwhile, gated recurrent units are incorporated to iteratively search for optimal matches on correlation maps, thereby enhancing the robustness of the system. Additionally, Gauss–Newton updates are embedded in iterations to impose geometrical constraints, which ensure accurate pose estimation. After end-to-end training on ScanNet, SVR-Net is evaluated on TUM-RGBD and successfully estimates poses on all nine scenes, while traditional ORB-SLAM fails on most of them. Furthermore, absolute trajectory error (ATE) results demonstrate that the tracking accuracy is comparable to that of DeepV2D. Unlike most previous monocular SLAM systems, SVR-Net directly estimates dense TSDF maps suitable for downstream tasks with high efficiency of data exploitation. This study contributes to the development of robust monocular visual SLAM systems and direct TSDF mapping.

Keywords: monocular SLAM; deep learning; TSDF mapping

check for updates

Citation: Lang, R.; Fan, Y.; Chang, Q. SVR-Net: A Sparse Voxelized Recurrent Network for Robust Monocular SLAM with Direct TSDF Mapping. *Sensors* **2023**, *23*, 3942. https://doi.org/10.3390/s23083942

Academic Editors: Teng Huang, Qiong Wang and Yan Pang

Received: 9 March 2023
Revised: 5 April 2023
Accepted: 9 April 2023
Published: 13 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A simultaneous localization and mapping (SLAM) system localizes the agent in the environment and reconstructs the environment. It is fundamental to other tasks, such as navigation and planning, etc. In visual SLAM, the agent senses the environment using a monocular, stereo or RGBD camera, and monocular SLAM is the most challenging due to difficulties in robust pose estimation and map construction. Moreover, the demand for dense maps in many applications presents a new set of challenges that must be addressed by existing monocular SLAM methods.

Typically, the pose of the camera is represented by a quantity in the special Euclidean group SE(3), which includes components of translation and rotation. The map in SLAM is often represented by a depth map, which contains the distance between each environmental point corresponding to each pixel in the image and the camera. Traditional SLAM methods are classed as either direct or indirect, according to the way they use information. Direct methods solve for poses and depths through photometric error. Photometric error utilizes abundant information such as lines and intensity variations, but complex patterns of images can cause local minima when optimizing photometric error. The optimization process is also unstable when an image is noisy. Indirect approaches exploit information from feature points. Feature points locate sparse local regions of images and compute descriptors which are similar from different perspectives. Reprojection error is optimized to minimize the distance of projection and matched location of feature points. For example, ORB-SLAM [1] uses a sparse ORB feature for tracking and mapping, which can achieve high performance

when feature points are correctly matched. However, in regions with few or repetitive textures, feature points are often erroneously detected or matched, which also leads to a robustness problem.

With the rapid growth of deep learning, neural networks have been successfully applied in image processing due to their powerful feature extraction capability [2,3]. They have also been adopted in SLAM to tackle the robustness problem. Some work has studied replacing submodules of SLAM with networks. In high-level feature extraction, using networks can provide an advantage in computing feature points [4–8] or finding good correspondences [9,10]. Learning-based outlier rejection is explored to produce robust matches [11–13]. Other work has investigated performing end-to-end learned SLAM [14–17]. After training on a wide variety of scenes with ground-truth poses, the networks can utilize prior information to avoid catastrophic errors.

However, previous end-to-end SLAM networks are not trained on applicable dense maps, which limits their availability. Networks such as TartanVo [16] and DROID-SLAM [15] only provide pose results. Other networks including DeepV2D [14] use depth maps or point clouds as map representation, which is not enough for downstream tasks. Post-processing and fusion could directly use depth maps or point clouds to produce maps suitable for downstream tasks, but additional information extracted from the networks would be omitted.

Instead of depth maps or point clouds, denser representation of environment geometry is considered, which is typically required to enable downstream tasks. To reconstruct a dense scene from a set of calibrated views, traditional multi-view stereo techniques rely on local depth map computation and global depth map fusion. The Truncated Signed Distance Function (TSDF) is a fusion format that contains distance information useable for navigation and planning. TSDF assigns each 3D point a signed distance from the nearest environment surface. The sign is decided by whether the point is hidden by the surface, and distances with absolute values greater than 1 are truncated. The result of two-stage reconstruction is prone to being either layered or scattered due to depth inconsistency [18]. Thus, deep neural architectures regressing TSDF for direct 3D reconstruction were designed in recent studies [18–20]. However, these methods require the camera poses as prior information, and they cannot guarantee the quality of maps when pose estimation is not accurate. In monocular SLAM, map construction is more challenging because there are more uncertainties in pose estimation.

In this work, we propose a monocular SLAM system with direct TSDF mapping based on our sparse voxelized recurrent network, SVR-Net. For a sequence of monocular image frames, SVR-Net outputs both poses and a TSDF map. The network takes an end-to-end pipeline that extracts features for correlation and recursively matches them, which is shown to have STOA localization performance in DROID-SLAM. Moreover, to enable TSDF reconstruction in the SLAM pipeline, a novel matching network is designed using sparse 3D convolution on voxels, combining the recursive matching with the TSDF mapping branch. This study makes a contribution towards the advancement of robust monocular visual SLAM systems and direct TSDF mapping.

The main features of the proposed method are summarized as follows:

(1) Robust monocular SLAM. SVR-Net's semantic encoder encodes information such as the scale of scenes to guide matching, which helps monocular pose estimation. SVR-Net utilizes correlation operations to reduce both the size of features and dependence on a specific scene's semantic information, which avoids overfitting. After end-to-end training on ScanNet [21] with ground truth poses and maps, SVR-Net successfully estimates poses for all nine scenes of the challenging TUM-RGBD [22] benchmark, whereas ORB-SLAM fails for most of them.

(2) Accurate pose estimation. Iterative updates are carried out in a recurrent network to search for the optimal match. Gauss–Newton updates are embedded in iterations to impose geometrical constraints, which improves the accuracy of pose estimation.

Experimental results using the TUM-RGBD benchmark show that the pose accuracy of SVR-Net is comparable to that of DeepV2D.

(3) Direct TSDF mapping. SVR-Net directly regresses the TSDF values and occupancy confidences of given voxels. Unlike previous monocular SLAM systems that produce depth maps or sparse 3D points, SVR-Net produces dense TSDF maps suitable for downstream tasks including navigation and planning. Compared with TSDF reconstruction methods that take depth maps as intermediate representation, direct TSDF mapping avoids depth inconsistency. Moreover, SVR-Net's direct TSDF mapping is more data-efficient because both pose and map are estimated using the same features.

2. Related Works

Monocular SLAM uses monocular visual information to locate the carrier and model the environment. Traditional methods are classified as either direct or indirect methods according to the way information is used [23]. Indirect approaches first detect and match key points between two frames, then predict poses and 3D points by minimizing the reprojection error [1,24]. Direct approaches operate directly on pixel intensities, and they estimate poses and depths through photometric error [23,25,26]. However, both direct and indirect methods are vulnerable to outliers.

Deep learning has been proposed to improve the robustness of SLAM systems. Some work has investigated replacing hand-crafted with learned features [4–8], using neural 3D representations [27–33], and combining learned energy terms with classical optimization backends [34,35]. In other work, researchers have tried to produce end-to-end learning for SLAM or VO systems [14–17]. Among these methods, DROID-SLAM consists of end-to-end recurrent iterative updates of camera pose and pixel-wise depth. It achieves large improvements in accuracy and substantially reduces catastrophic failures. Sparse voxel-based networks have been applied for the efficient processing of point clouds and voxels [36–38]. SVR-Net contains an iterative pipeline that is similar to DROID-SLAM, but it estimates TSDFs, instead of depth values, with a sparse voxelized structure for direct mapping.

The implicit representations of surface geometry through SDF are introduced in [39]. The efficacy of TSDFs in fusing high-rate, noisy depth data from consumer-grade depth cameras, which was first demonstrated by Newcombe et al. [40], has led to a rapid rise in their popularity. Furthermore, this representation has led to their recent adoption on robotic platforms [41–43], where the distance field has shown additional utility for optimization-based motion planning [44,45].

Recent studies have shown the advantage of deep networks in TSDF mapping, compared with traditional multi-view methods. NeuralRecon directly reconstructs local surfaces, represented as sparse TSDF volumes for each video fragment sequentially, by a neural network [18]. VoRTX is occlusion-aware, leveraging the transformer architecture to predict an initial, projective scene geometry estimate [20]. VolumeFusion replicates the traditional two-stage framework with deep neural networks, improving both the interpretability and the accuracy of the results [46]. The suggested network is closely related to TSDF mapping networks, but does not need camera poses to be provided. Table 1 shows a comparison of existing learning-based localization and mapping methods.

Table 1. Comparison of existing learning-based approaches.

References	Year	Method	Monocular Input	Pose Estimation	TSDF Mapping
[28]	2020	DeepFactors	Yes	Yes	No
[35]	2020	D3VO	Yes	Yes	No
[14]	2020	DeepV2D	Yes	Yes	No
[32]	2021	IMAP	No	Yes	No
[15]	2021	DROID-SLAM	Yes	Yes	No
[18]	2021	NeuralRecon	Yes	No	Yes
[20]	2021	VoRTX	Yes	No	Yes
[46]	2021	VolumeFusion	Yes	No	Yes
[33]	2022	NICE-SLAM	No	Yes	Yes

3. Method

The SLAM system employs a coarse-to-fine strategy to achieve tracking and global dense mapping, as shown in Figure 1. At the first stage, the raw pose and local map of a pair of input frames are estimated using SVR-Net. The map is represented by sparse voxels with TSDF values. Then, the map is fused with the first-stage global map. At the second stage, the voxels are up-sampled, halving their intervals. Subsequently, the same network is utilized to refine the pose and map. The resulting fine local map is fused with the second-stage global map, and SVR-Net proceeds to track the next pair of frames.

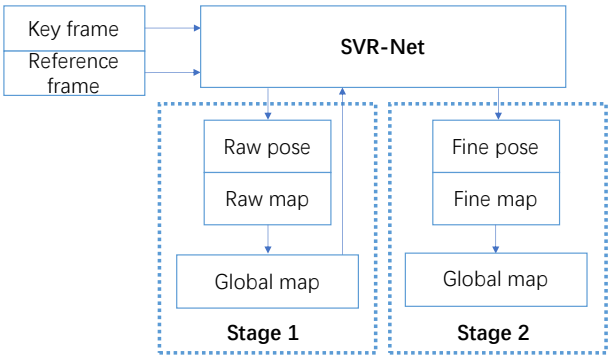

Figure 1. SLAM system.

SVR-Net is an end-to-end network for monocular simultaneous tracking and mapping. Its input consists of a pair of RGB frames with a query set of voxel coordinates. It outputs the relative pose $\boldsymbol{T}_n$ of the frames and TSDF values $\boldsymbol{d}_n$ of the voxels. For each pair of frames, the earlier frame is designated as the keyframe, and the later frame is designated as the reference frame. As shown in Figure 2, SVR-Net first extracts the image features into 2D feature maps. The feature map of the key frame is then transformed into feature voxels and correlated with the features of the reference frame. After sampling based on the current pose estimation, a matching network iteratively matches the features and updates the estimation of pose and map. The iterative pipeline performs a search in a correlation field of the features to find the optimal match, thereby enhancing both accuracy and robustness.

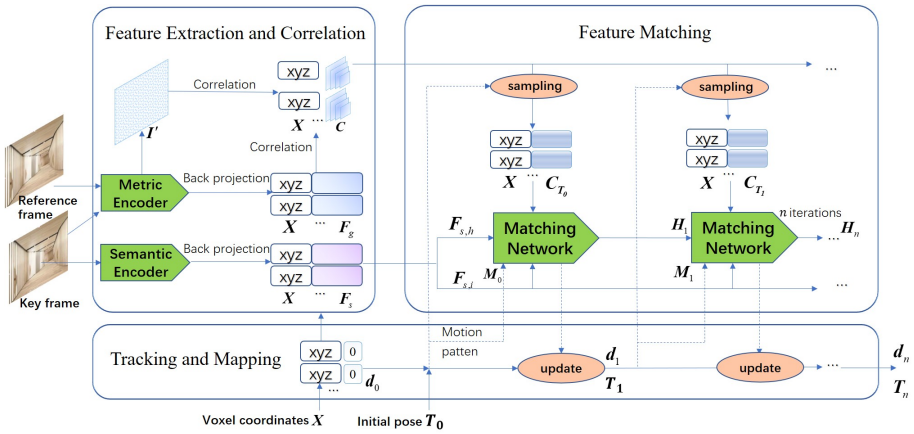

Figure 2. The structure of SVR-Net.

3.1. Voxel Feature Extraction and Correlation

In the feature extraction, deep features of the key frame are computed and back-projected into 3D voxels. Then, correlation maps between features of the voxels and the reference frame are computed to provide similarity information for matching.

3.1.1. Voxel Feature Extraction

2D feature maps are extracted from the input images using 2D convolution networks, namely, a metric encoder and a semantic encoder. Both networks produce feature maps at 1/8 the input image resolution. The metric encoder is a subnet of MnasNet [47] and contains two inverted residual blocks. The metric features of both frames are used to build correlation between voxels and images. The semantic encoder resembles the context network in RAFT, which consists of six residual blocks. The reference frame is only fed into the metric encoder to produce a metric map $\boldsymbol{I}_g^{'}$ for feature correlation, while the semantic features of the key frame provide the initial hidden state and inputs for recurrent feature matching.

Given voxel coordinates, both the semantic and metric feature maps of the key frame are back-projected to produce the 3D feature volume. Assume that there are N input voxels with their coordinates $\boldsymbol{X}_i, 1 \leq i \leq N$. For each voxel coordinate, let the projected coordinate be $\boldsymbol{x}_i$.

$$\boldsymbol{x}_i = \Pi(\boldsymbol{T}_k \boldsymbol{X}_i) \tag{1}$$

Here, Π is the projection function of the camera that maps 3D points onto the image. $\boldsymbol{T}_k$ is the transformation matrix of the key frame. Bilinear interpolation is carried out at $\boldsymbol{x}_i$ on the feature maps of the key frame to produce voxelized features of $\boldsymbol{X}_i$. To exploit sparsity, the coordinates and features are stored in separate matrices $\boldsymbol{X}$, $\boldsymbol{F}_s$ and $\boldsymbol{F}_g$ with N rows. Here, the subscript s and g represent semantic features and metric features, respectively.

3.1.2. Voxel Feature Correlation

After the voxelization of features from the key frame, the feature correlation between the key frame and the reference frame becomes feature correlation between the feature voxels and the reference frame's metric feature map. The correlation maps are computed with dot products within the metric features. For the ith voxel, its correlation map at the first layer is $\boldsymbol{C}_i^1$.

$$\boldsymbol{C}_i^1 \in \mathbb{R}^{h \times w}, C_{ikl}^1 = \sum_{j=1}^{D} F_{g,ij} I_{g,klj}^{'} \tag{2}$$

Here, h, w, and d are the height, width, and feature dimensions of the metric feature map. $F_{g,ij}$ is the jth value of the voxel's metric feature. $I_{g,klj}^{'}$ is the jth value of the reference

frame's metric feature at position (k, l). The feature correlation reduces feature dimensions and provides scene-independent information to the downstream networks, which can improve generalization.

To present similarity information at different scales, the $\boldsymbol{C}_i^1$ is pooled with kernel size 2,4,8. All generated maps are $\boldsymbol{C}, \boldsymbol{C} = \{\boldsymbol{C}_i^m, 1 \leq i \leq N, 1 \leq m \leq 4\}$. $\boldsymbol{C}_i^m$ has a resolution of $\frac{h}{2^{m-1}} \times \frac{w}{2^{m-1}}, m = 1, 2, 3, 4$. The four layers of the correlation maps form a pyramid where the correlation field is sampled at each iteration.

3.2. Sparse Recurrent Feature Matching

Feature matching uses an iterative pipeline to estimate the correspondence between the voxels and the reference frame. The voxels are first projected to the reference frame using the estimated pose from the last iteration. Then, correlation values sampled from projection coordinates are fed into the matching network, which predicts both updated TSDF values and corrected projection coordinates.

3.2.1. Sampling

In the sampling operation, correlation vectors are generated for all voxels and their projection coordinates on correlation maps, as demonstrated in Figure 3. For each correlation map of each projection coordinates $\boldsymbol{x_i}$, a grid with radius r is applied. Every point of the grid corresponds to an offset from the coordinate where the correlation value is sampled with bilinear interpolation. For all four correlation maps of the voxel, the values from four grids are concatenated to generate the correlation vector. All correlation vectors are stacked to a matrix $\boldsymbol{C_T} \in \mathbb{R}^{N \times 4(2r+1)^2}$. The subscript $\boldsymbol{T}$, which represents the pose of the reference frame, varies during iteration and results in different correlation values.

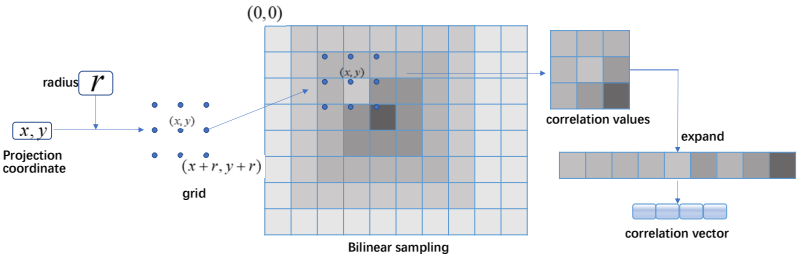

Figure 3. Sampling pipeline.

3.2.2. Matching Network

The coordinates and features of voxels stored in N-row matrices are fed into the matching network. The network produces three outputs: (1) revisions of voxels' TSDF values $\Delta \boldsymbol{d} \in \mathbb{R}^N$; (2) corrections of projection coordinates $\Delta \boldsymbol{x} \in \mathbb{R}^{N \times 2}$; and (3) confidence weights $\boldsymbol{w} \in \mathbb{R}_+^{N \times 3}$. Each row of the outputs corresponds to each input voxel. The first two columns of confidence weights represent the confidence of the projection coordinates. The last column is the confidence of voxel occupancy, whose value is between 0 and 1.

The matching network is implemented with sparse 3D convolution to process voxels with varying spatial distributions and sparsity. A gated recurrent unit (GRU) is also adopted to support iteration. The formulation of a GRU layer is as follows.

$$\begin{aligned} \boldsymbol{Z}_t &= \sigma(SConv_3([\boldsymbol{H}_{t-1}, \boldsymbol{G}_t])) \\ \boldsymbol{R}_t &= \sigma(SConv_3([\boldsymbol{H}_{t-1}, \boldsymbol{G}_t])) \\ \widetilde{\boldsymbol{H}}_t &= \tanh(SConv_3(\boldsymbol{R}_t \odot \boldsymbol{G}_t)) \\ \boldsymbol{H}_t &= (\boldsymbol{1} - \boldsymbol{Z}_t) \odot \boldsymbol{H}_{t-1} + \boldsymbol{Z}_t \odot \widetilde{\boldsymbol{H}}_t \end{aligned} \tag{3}$$

Here, $\odot$ is element-wise multiplication and σ is sigmoid activation. Sparse 3D convolution, $SConv_3$, implicitly utilizes voxel coordinates to fuse features. The hidden state from the last iteration is denoted by $\boldsymbol{H}_{t-1}$. The semantic feature $\boldsymbol{F}_s$ is split into $\boldsymbol{F}_{s,h}$ and $\boldsymbol{F}_{s,i}$, to provide the initial hidden state and input feature, respectively. The input feature

$\boldsymbol{G}_t$ of the GRU is a concatenation of these features: (1) the semantic feature for input $\boldsymbol{F}_{s,i}$; (2) a correlation feature $\boldsymbol{F}_c$; and (3) a motion feature $\boldsymbol{F}_m$. The correlation feature is obtained by encoding $\boldsymbol{C_T}$ with two $SConv_3$ layers. The motion feature is obtained by encoding a motion pattern matrix $\boldsymbol{M}$ with a similar network, which is informative for GRU to perceive its progress. The motion pattern consists of flows and errors of projection coordinates, as demonstrated in Section 3.3.

The remaining components of the matching network are three branches to produce outputs at iteration t: $\Delta \boldsymbol{d}_t$, $\Delta \boldsymbol{x}_t$ and $\boldsymbol{w}_t$. Each branch contains two $SConv_3$ layers and takes $\boldsymbol{H}_t$ as common input, which effectively utilizes information in features.

3.3. Iterative Tracking and Mapping

The outputs of feature matching are utilized to update the estimations of the pose and map in each iteration of SVR-Net. A local map is obtained from the final iteration. Then, map fusion of the SLAM pipeline integrates the local maps from all input frames to enhance global consistency.

3.3.1. Pose Estimation with Gauss–Newton Update

After each iteration of feature matching, the estimated projection coordinates from the last iteration are corrected with $\Delta \boldsymbol{x}$. Then, under the constraints of the projection coordinates and the confidence weights, the classical re-projection error E is optimized using Gauss–Newton iteration to produce an updated pose estimation.

$$E = \sum_{i=1}^{N} \boldsymbol{e}_i^T \boldsymbol{W}_i \boldsymbol{e}_i \tag{4}$$

Here, $\boldsymbol{e}_i$ is the re-projection error from the ith voxel. $\boldsymbol{W}_i$ is the corresponding weight matrix based on the confidence weights $\boldsymbol{w}$.

$$\begin{aligned} \boldsymbol{e}_i &= \boldsymbol{x}_i + \Delta \boldsymbol{x}_i - \Pi(\boldsymbol{T}\boldsymbol{X}_i) \\ \boldsymbol{W}_i &= diag(w_{i1}, w_{i2}) \end{aligned} \tag{5}$$

The Gauss–Newton update on the pose $\boldsymbol{T}$ is performed within Lie algebra. The update is differential with respect to $\Delta \boldsymbol{x}$, making it possible to train the entire network end-to-end.

The motion pattern of the updated pose is informative to the matching network because the network can learn to adjust its matching strategy reactively at each iteration. The motion pattern $\boldsymbol{M}$ contains the flows and errors of all voxels. Let the initial pose before the first iteration be $\boldsymbol{T}_0$ and the estimated pose at iteration t be $\boldsymbol{T}_t$. Then, the flow of the ith voxel is defined as $\Pi(\boldsymbol{T}_t \boldsymbol{X}_i) - \Pi(\boldsymbol{T}_0 \boldsymbol{X}_i)$, which informs the corresponding optical flow on the image. The error is the re-projection error of $\boldsymbol{T}_t$.

3.3.2. Map Update and Fusion

For each pair of frames, a local map within the key frame's view frustum is estimated. The map is represented as a set of voxels with TSDF values and occupancy confidences. The coordinates of the voxels are initialized, filling the view frustum with a maximum depth. The distance between neighboring coordinates, i.e., voxel size, controls the resolution of the map. However, when the voxel size is small, the huge number of voxels can cause a heavy memory burden on the GPU. This problem is addressed using the coarse-to-fine strategy.

At the first stage, the TSDF values $\boldsymbol{d}_0$ in the view frustum are initialized with zero. With iterative running of the matching network, the TSDF values are successively added with revisions $\Delta \boldsymbol{d}$. After the final iteration of the network, the first-stage TSDF estimations are obtained with the confidence weights $\boldsymbol{w}$. Occupancy confidences $\boldsymbol{w}_o$ are the last column of the weights, which are used in map fusion. After the first-stage map fusion, the voxels in the view frustum are up-sampled and fed into SVR-Net again for second-stage estimation, producing a fine-grained map.

As the frames are input in sequence in the SLAM pipeline, all local maps are fused together to produce a global map. The global map is computed using a linear weighting method similar to TSDF integration [40]. The global map is initialized with the local map of the first key frame. For each subsequent local map, the global TSDF values of overlapped voxels are averaged according to global and local occupancy confidences. Then, the global occupancy confidences are updated in a cumulative manner. The data of voxels in a new region are directly attached to the global map.

4. Experiments

SVR-Net is trained on ScanNet(V2), and the full SLAM system is evaluated using TUM-RGBD. The ScanNet dataset includes 1613 indoor scenes with ground-truth camera poses and depth maps. The TUM-RGBD dataset contains nine indoor scenes with ground-truth poses. Without depth maps, the RGB images in TUM-RGBD contain heavy motion blur, which is a challenge for monocular SLAM.

ScanNet is split into a training set and a validation set, with the training set containing 1513 scenes. In order to improve tracking accuracy with various motion patterns, each pair of training frames is sampled using a random selection strategy from a successive subsequence. The length of each subsequence is controlled to satisfy the condition that there are exactly nine frames with mutual distance greater than thresholds. The distance threshold of rotation is 15°, and the threshold of translation is 0.1 m. The ground-truth TSDF global maps are generated by integrating depth frames of the training sequences. For each sequence, there are two scales of ground-truth maps, corresponding to two stages. The voxel sizes of the two scales are 16 cm and 8 cm. For training, the ground-truth map of each image frame is set to the part of the global map within the frame's view frustum. Training supervision includes both pose loss and map loss. The pose loss measures the distance between ground truth and the predicted pose, $L_{pose} = \|Log_{SE3}(\boldsymbol{T}^{-1}\boldsymbol{T}_{gt})\|_2$. The map loss is the mean difference between the TSDF values, $L_{map} = \|\boldsymbol{d} - \boldsymbol{d}_{gt}\|_1$.

The hardware used for training and evaluation includes Intel Xeon Silver 4210R @2.4GHz, 32 GB RAM and Nvidia Quadro RTX6000. The software infrastructure includes CUDA10.1, Python, PyTorch, LieTorch and Open3D.

4.1. Results on Matching

The quality of feature matching influences the performance of both tracking and mapping. To validate the superiority of network-based feature matching, matching results from SVR-Net are compared with results from the traditional method. The results are presented in both 3D and 2D visualizations, and the quality of feature matching is quantified using the end-point-error (EPE) metric.

The visualization of feature matching in 3D form is shown in Figures 4 and 5. The feature matching is performed in a test scene of ScanNet. The blue-colored point clouds represent the central coordinates of voxels. Each voxel is implicitly attached with a feature vector obtained through voxel feature extraction. The black lines are match lines connecting voxels with their matched estimations of projection coordinates on the reference frame. Each pair of panels represents the same match from different perspectives. The left panel shows the pose of the reference frame and the RGB image represented by a point lattice. The right panel shows the result observed from the back of the reference frame, where the voxels in the environment almost overlap with their projection on the image.

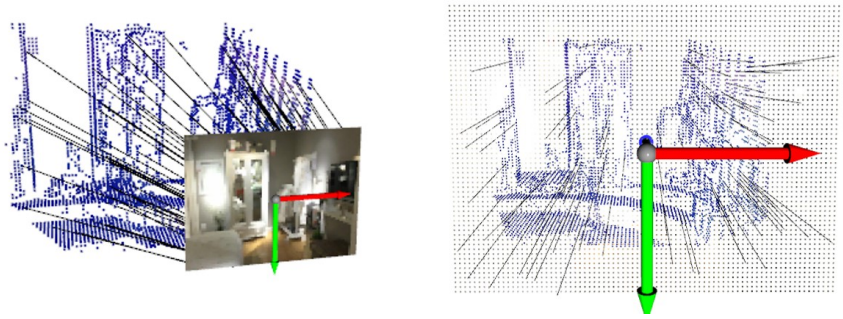

Figure 4. Matching result from SVR-Net(3D).

As illustrated in Figure 4, the match lines of SVR-Net are consistent and conform to the projection rule, where voxels representing an object (such as the wardrobe) are projected onto the corresponding object in the image. The match lines in the right panel have short projection lengths, indicating low end-point-error. This is attributed to the integration capability of the matching network for the features of adjacent voxels.

Figure 5 illustrates match lines obtained through brute-force search, which contain inconsistent parts that indicate outliers. Some of the match lines in the right figure exhibit significant projection length with high end-point-error. The result indicates that estimation from SVR-Net conforms to projection rules, while the estimation of a feature's L2 distance in brute-force search has no constraints from adjacent voxels, and this method is prone to producing outliers. A conventional ratio test can remove outliers, but reduces the available matches at the same time, as shown in the following Figures 6 and 7. All frames in the Figures are included in ScanNet test scenes.

As illustrated in Figure 6, SVR-Net produces dense matches between 2D image frames. To ensure clarity, only the top 200 matches from SVR-Net, possessing the highest confidence weights, are represented. There are no outliers in the matches, and correct matches can be obtained even in low-texture regions. It can be observed that SVR-Net identified the lines in the scene through high-confidence matches. In the matching network, the position information of the lines is propagated to adjacent low-texture regions, enabling the network to match correctly in these areas and improving its robustness.

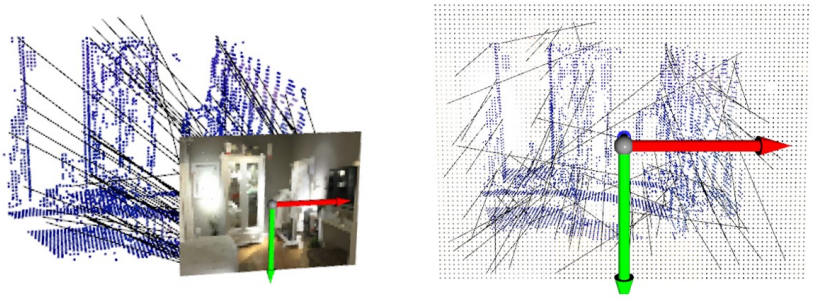

Figure 5. Matching result from brute-force search(3D).

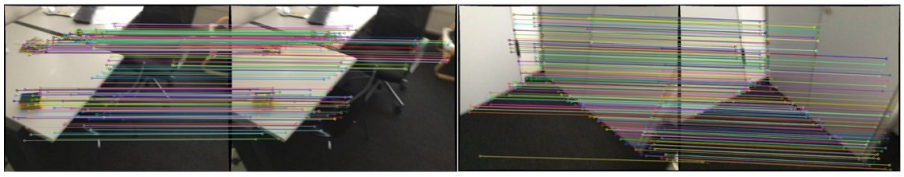

Figure 6. Matching result from SVR-Net(2D).

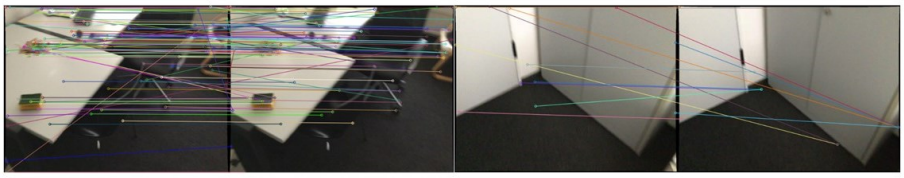

Figure 7. Matching result from SIFT(2D).

For comparison, Figure 7 illustrates the classical approach, which involves the extraction of SIFT feature points and their matches through brute-force search. To eliminate outliers, only matches with a test ratio greater than 0.9 from the classical approach are retained. However, some outliers still remain, due to repetitive features in the low-texture regions, which interfere with pose estimation and reduce robustness. Furthermore, the sparse matches preclude the generation of a dense map.

EPE is computed to evaluate the quality of feature matching. For a set of k feature points in a key frame, we denote their estimated matching coordinates at the corresponding reference frame as $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_k$, and their true matching coordinates as $\widehat{\boldsymbol{x}}_1, \widehat{\boldsymbol{x}}_2, \ldots, \widehat{\boldsymbol{x}}_k$. Then, the end-point-error is

$$\frac{1}{k}\sum_{i=1}^{k}\|\boldsymbol{x}_i - \widehat{\boldsymbol{x}}_i\|_2. \tag{6}$$

The EPE of a varying number of feature points is plotted in Figure 8. The evaluation involves 304 pairs of frames in two test scenes from the ScanNet dataset. In order to provide a given number of matched features, only matches with the highest confidence weights or test ratios are selected. As the number of matches increases, traditional SIFT-based methods become less effective in distinguishing repeated features, resulting in an increase in EPE. In contrast, SVR-Net experiences a decrease in EPE as the measurement quantity increases. When there are numerous matches, the EPE of SVR-Net is significantly lower than that of traditional methods, indicating that it is more robust for dense matching.

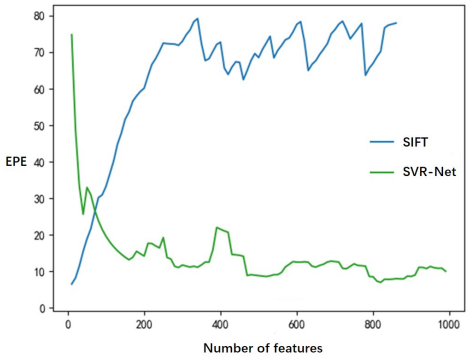

Figure 8. EPE at different numbers of feature points.

4.2. Results On Full System

SVR-Net estimates both raw and fine partial maps for every pair of adjacent frames. The partial dense TSDF maps of a test scene in ScanNet are shown in Figure 9. The maps in (a) and (b) are estimated maps with voxel sizes of 16 cm and 8 cm, and the map in (c) is the ground-truth map with a voxel size of 4 cm. The RGB image in (d) is the corresponding key frame.

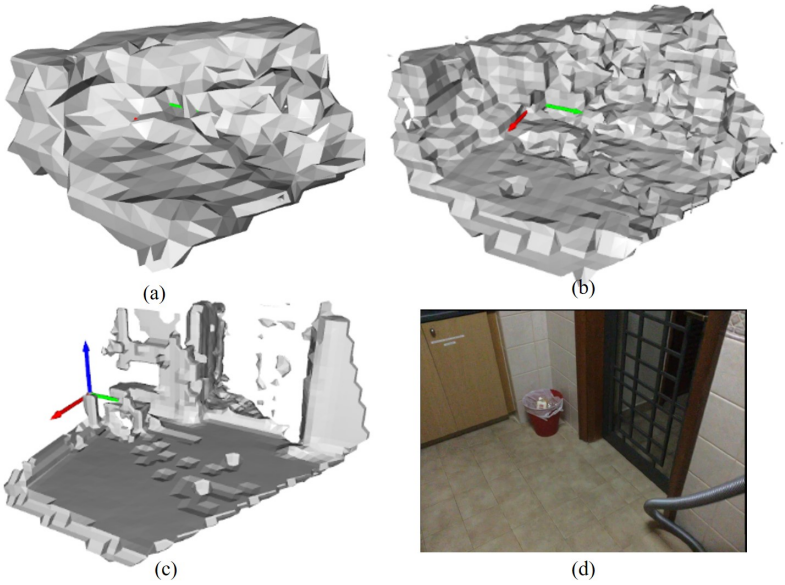

Figure 9. Partial dense TSDF maps and corresponding image. Estimated maps with voxel sizes of 16 cm and 8 cm are shown in (**a**,**b**) respectively. The ground-truth map with a voxel size of 4 cm is shown in (**c**). The RGB image from the corresponding key frame is in (**d**).

The full SLAM results on scene 360 of the TUM-RGBD dataset are shown in Figure 10. Camera trajectory and global maps are processed incrementally in the SLAM system. The estimated trajectory is displayed in blue, while the ground truth is in black. The dense global map is showcased by both the TSDF map and the point cloud of voxel coordinates at a voxel size of 8 cm. It can be observed that the pose estimated by the network almost coincides with the ground truth pose. The TSDF map in the left panel illustrates the floor and wall of the environment. The right panel presents the distribution of environment points through a dense point cloud.

The monocular SLAM system is evaluated on nine indoor scenes from the TUM-RGBD dataset. For each scene, the system runs on sequential RGB frames and outputs the result of tracking and dense mapping. The tracking performance is compared with those of ORB-SLAM and DeepV2D. All trajectories are evaluated up to scale using absolute trajectory error (ATE), which computes the rooted mean square translation error between estimated and ground truth poses. The result is shown in Table 2. All methods are provided mono. In challenging scenarios, the conventional ORB SLAMs fail, whereas deep approaches exhibit greater robustness, effectively computing poses in all scenes. Furthermore, the SVR-Net model outperforms DeepV2D with respect to average ATE. It is noteworthy that, in situations where ORB SLAMs achieve success, learning-based methods are less accurate due to the generalization error of networks. However, in the context of learning-based approaches, SVR-Net, which only estimates the relative pose between adjacent frames, achieves a similar ATE to that of DeepV2D, which utilizes multi-frame joint optimization. This suggests that SVR-Net demonstrates lower generalization error.

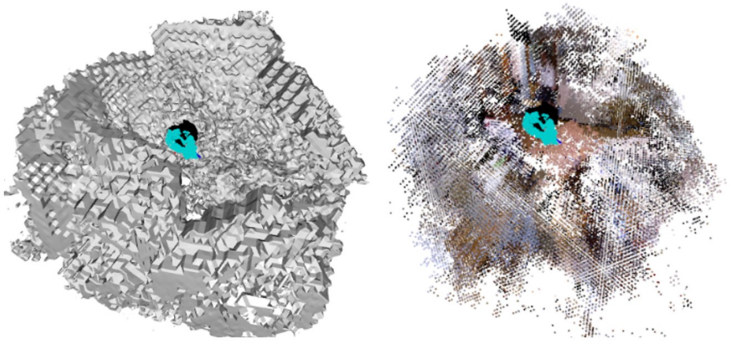

Figure 10. Full SLAM result.

Table 2. ATE on the TUM-RGBD benchmark. Failures are recorded as X.

Methods	360	Desk	Desk2	Floor	Plant	Room	rpy	Teddy	xyz	Average
ORB-SLAM2	X	0.071	X	0.023	X	X	X	X	0.01	-
ORB-SLAM3	X	0.017	0.21	X	0.034	X	X	X	0.009	-
DeepV2D	0.243	0.166	0.379	1.653	0.203	0.246	0.105	0.316	0.064	0.375
Ours	0.205	0.266	0.255	0.433	0.383	0.564	0.521	0.541	0.13	0.366

5. Discussion

Deep learning methods have been applied to the processes of monocular SLAM or dense TSDF mapping in various systems. However, there have been a limited number of studies integrating both of these aspects in a unified network. The suggested system, which is capable of simultaneously estimating pose and TSDF map using an end-to-end network, is more suitable for applications that require both aspects. The same features are exploited to produce two modalities of outputs with accuracy comparable to that of DeepV2D, which indicates highly efficient data utilization.

Real-time performance is an important metric for SLAM. SVR-Net runs at a frame rate of 4Hz on a desktop RTX2080. However, its performance on embedded devices or decentralized systems has not been tested, and this requires further research in the future.

SVR-Net only uses information from adjacent frames, which constrains the accuracy of localization and propagates errors to the map, thereby reducing its precision. A pose graph can significantly reduce pose error using multi-frame joint optimization. Additional research is necessary in the future to incorporate this into the proposed network.

6. Conclusions

This paper proposes SVR-Net, an end-to-end network for monocular TSDF SLAM. SVR-Net produces both poses and global TSDF maps from a sequence of monocular images. A metric encoder is constructed for the purpose of measuring image similarities, while a semantic encoder is devised to encode information, including the scale of scenes, to facilitate matching. In the matching network, sparse 3D convolution is adopted for dense TSDF mapping. A GRU is also employed to enable iterative updates for rectifying matches. Matching results presented in both 3D and 2D visualizations demonstrate that this approach effectively eliminates outliers, producing matches which are dense and more consistent than those produced by the traditional SIFT-based matching method. An experiment using the TUM-RGBD dataset further validates the conclusion that SVR-Net exhibits greater robustness compared with traditional ORB SLAMs. SVR-Net integrates Gauss–Newton updates for accurate pose estimation. Our experiments demonstrate that SVR-Net achieves a level of localization accuracy comparable to that of DeepV2D. Unlike previous monocular SLAM methods, the network directly yields a dense TSDF map during localization, which avoids inconsistency from depth map fusion and is more feasible for downstream tasks that are highly dependent on dense maps. Compared with existing TSDF mapping methods, SVR-Net does not require pre-calibrated poses and can meet real-time requirements.

Author Contributions: Conceptualization, Y.F.; Methodology, Y.F.; Software, Y.F.; Resources, R.L.; Writing—original draft, Y.F.; Writing—review & editing, R.L.; Supervision, Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; M. Montiel, J.M.; D. Tardós, J. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
2. Suryanarayana, G.; Chandran, K.; Khalaf, O.I.; Alotaibi, Y.; Alsufyani, A.; Alghamdi, S.A. Accurate Magnetic Resonance Image Super-Resolution Using Deep Networks and Gaussian Filtering in the Stationary Wavelet Domain. *IEEE Access* **2021**, *9*, 71406–71417. [CrossRef]
3. Yue, Z.; Gao, F.; Xiong, Q.; Wang, J.; Huang, T.; Yang, E.; Zhou, H. A Novel Semi-Supervised Convolutional Neural Network Method for Synthetic Aperture Radar Image Recognition. *Cogn. Comput.* **2021**, *13*, 795–806. [CrossRef]
4. Choy, C.; Gwak, J.; Savarese, S.; Chandraker, M. Universal Correspondence Network. *arXiv* **2016**, arXiv:1606.03558.
5. DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperPoint: Self-Supervised Interest Point Detection and Description. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 337–33712. [CrossRef]
6. Luo, Z.; Shen, T.; Zhou, L.; Zhu, S.; Zhang, R.; Yao, Y.; Fang, T.; Quan, L. GeoDesc: Learning Local Descriptors by Integrating Geometry Constraints. In Proceedings of the Computer Vision—ECCV, Munich, Germany, 8–14 September 2018; Lecture Notes in Computer Science; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 170–185. [CrossRef]
7. Mishchuk, A.; Mishkin, D.; Radenović, F.; Matas, J. Working Hard to Know Your Neighbor's Margins: Local Descriptor Learning Loss. *arXiv* **2017**, arXiv:1705.10872.
8. Ono, Y.; Trulls, E.; Fua, P.; Yi, K.M. LF-Net: Learning Local Features from Images. In Proceedings of the Advances in Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; Curran Associates, Inc.: Red Hook, NY, USA, 2018; Volume 31.
9. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning Feature Matching with Graph Neural Networks. *arXiv* **2020**, arXiv:1911.11763.
10. Yi, K.M.; Trulls, E.; Ono, Y.; Lepetit, V.; Salzmann, M.; Fua, P. Learning to Find Good Correspondences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2666–2674.
11. Brachmann, E.; Krull, A.; Nowozin, S.; Shotton, J.; Michel, F.; Gumhold, S.; Rother, C. DSAC — Differentiable RANSAC for Camera Localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2492–2500. [CrossRef]
12. Brachmann, E.; Rother, C. Neural-Guided RANSAC: Learning Where to Sample Model Hypotheses. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4322–4331.
13. Kluger, F.; Brachmann, E.; Ackermann, H.; Rother, C.; Yang, M.Y.; Rosenhahn, B. CONSAC: Robust Multi-Model Fitting by Conditional Sample Consensus. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4633–4642. [CrossRef]
14. Teed, Z.; Deng, J. DeepV2D: Video to Depth with Differentiable Structure from Motion. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
15. Teed, Z.; Deng, J. DROID-SLAM: Deep Visual SLAM for Monocular, Stereo, and RGB-D Cameras. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 6–14 December 2021; Curran Associates, Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 16558–16569.
16. Wang, W.; Hu, Y.; Scherer, S. TartanVO: A Generalizable Learning-based VO. In Proceedings of the 2020 Conference on Robot Learning (PMLR), London, UK, 8–11 November 2020; pp. 1761–1772.
17. Zhou, H.; Ummenhofer, B.; Brox, T. DeepTAM: Deep Tracking and Mapping. In *Proceedings of the Computer Vision–ECCV 2018*; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; pp. 851–868. [CrossRef]
18. Sun, J.; Xie, Y.; Chen, L.; Zhou, X.; Bao, H. NeuralRecon: Real-Time Coherent 3D Reconstruction from Monocular Video. *arXiv* **2021**, arXiv:2104.00681.
19. Murez, Z.; As, T.; Bartolozzi, J.; Sinha, A.; Badrinarayanan, V.; Rabinovich, A. Atlas: End-to-End 3D Scene Reconstruction from Posed Images. *arXiv* **2020**, arXiv:2003.10432.

20. Stier, N.; Rich, A.; Sen, P.; Höllerer, T. VoRTX: Volumetric 3D Reconstruction with Transformers for Voxelwise View Selection and Fusion. In Proceedings of the 2021 International Conference on 3D Vision (3DV), London, UK, 1–3 December 2021; pp. 320–330. [CrossRef]
21. Dai, A.; Chang, A.X.; Savva, M.; Halber, M.; Funkhouser, T.; Nießner, M. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2432–2443. [CrossRef]
22. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 573–580. [CrossRef]
23. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
24. Leutenegger, S.; Furgale, P.; Rabaud, V.; Chli, M.; Konolige, K.; Siegwart, R. *Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization*; ETH Library: Zurich, Switzerland, 2013. [CrossRef]
25. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. In *Proceedings of the Computer Vision–ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2014; pp. 834–849. [CrossRef]
26. Ferrera, M.; Eudes, A.; Moras, J.; Sanfourche, M.; Le Besnerais, G. OV^{2}SLAM: A Fully Online and Versatile Visual SLAM for Real-Time Applications. *IEEE Robot. Autom. Lett.* **2021**, *6*, 1399–1406. [CrossRef]
27. Bloesch, M.; Czarnowski, J.; Clark, R.; Leutenegger, S.; Davison, A.J. CodeSLAM—Learning a Compact, Optimisable Representation for Dense Visual SLAM. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2560–2568.
28. Czarnowski, J.; Laidlow, T.; Clark, R.; Davison, A.J. DeepFactors: Real-Time Probabilistic Dense Monocular SLAM. *IEEE Robot. Autom. Lett.* **2020**, *5*, 721–728. [CrossRef]
29. Kopf, J.; Rong, X.; Huang, J.B. Robust Consistent Video Depth Estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Montreal, QC, Canada, 10–17 October 2021; pp. 1611–1621.
30. Luo, X.; Huang, J.B.; Szeliski, R.; Matzen, K.; Kopf, J. Consistent Video Depth Estimation. *ACM Trans. Graph.* **2020**, *39*, 71:1–71:13. [CrossRef]
31. Sucar, E.; Wada, K.; Davison, A. NodeSLAM: Neural Object Descriptors for Multi-View Shape Reconstruction. In Proceedings of the 2020 International Conference on 3D Vision (3DV), Fukuoka, Japan, 25–28 November 2020; pp. 949–958. [CrossRef]
32. Sucar, E.; Liu, S.; Ortiz, J.; Davison, A.J. iMAP: Implicit Mapping and Positioning in Real-Time. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 6229–6238.
33. Zhu, Z.; Peng, S.; Larsson, V.; Xu, W.; Bao, H.; Cui, Z.; Oswald, M.R.; Pollefeys, M. NICE-SLAM: Neural Implicit Scalable Encoding for SLAM. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; pp. 12776–12786. [CrossRef]
34. Yang, N.; Wang, R.; Stückler, J.; Cremers, D. Deep Virtual Stereo Odometry: Leveraging Deep Depth Prediction for Monocular Direct Sparse Odometry. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2018; pp. 835–852. [CrossRef]
35. Yang, N.; von Stumberg, L.; Wang, R.; Cremers, D. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 1278–1289. [CrossRef]
36. Tang, H.; Liu, Z.; Zhao, S.; Lin, Y.; Lin, J.; Wang, H.; Han, S. Searching Efficient 3D Architectures with Sparse Point-Voxel Convolution. In Proceedings of the Computer Vision—ECCV 2020, Glasgow, UK, 23–28 August 2020; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2020; pp. 685–702. [CrossRef]
37. Zhao, L.; Xu, S.; Liu, L.; Ming, D.; Tao, W. SVASeg: Sparse Voxel-Based Attention for 3D LiDAR Point Cloud Semantic Segmentation. *Remote Sens.* **2022**, *14*, 4471. [CrossRef]
38. Liu, L.; Gu, J.; Zaw Lin, K.; Chua, T.S.; Theobalt, C. Neural Sparse Voxel Fields. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 15651–15663.
39. Curless, B.; Levoy, M. A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of the SIGGRAPH96: 23rd International Conference on Computer Graphics and Interactive Techniques*; Association for Computing Machinery: New York, NY, USA, 1996. [CrossRef]
40. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Kim, D.; Davison, A.J.; Kohli, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. KinectFusion: Real-Time Dense Surface Mapping and Tracking. In Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 26–29 October 2011.
41. Lin, Y.; Gao, F.; Qin, T.; Gao, W.; Liu, T.; Wu, W.; Yang, Z.; Shen, S. Autonomous Aerial Navigation Using Monocular Visual-Inertial Fusion. *J. Field Robot.* **2018**, *35*, 23–51. [CrossRef]

42. Oleynikova, H.; Taylor, Z.; Fehr, M.; Siegwart, R.; Nieto, J. Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-Board MAV Planning. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canad, 24–28 September 2017; pp. 1366–1373. [CrossRef]
43. Wagner, R.; Frese, U.; Bäuml, B. Graph SLAM with Signed Distance Function Maps on a Humanoid Robot. In Proceedings of the 2014 IEEE/RSJ International Conference on ntelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2691–2698. [CrossRef]
44. Oleynikova, H.; Taylor, Z.; Siegwart, R.; Nieto, J. Safe Local Exploration for Replanning in Cluttered Unknown Environments for Microaerial Vehicles. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1474–1481. [CrossRef]
45. Ratliff, N.; Zucker, M.; Bagnell, J.A.; Srinivasa, S. CHOMP: Gradient Optimization Techniques for Efficient Motion Planning. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, 12–17 May 2009; pp. 489–494. [CrossRef]
46. Choe, J.; Im, S.; Rameau, F.; Kang, M.; Kweon, I.S. VolumeFusion: Deep Depth Fusion for 3D Scene Reconstruction. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 16066–16075. [CrossRef]
47. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MnasNet10-17 October 2021Platform-Aware Neural Architecture Search for Mobile. *arXiv* **2018**, arXiv:1807.11626.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.