applied sciences

MDPI

Article

Voxel-Based Scene Representation for Camera Pose Estimation of a Single RGB Image

Sangyoon Lee [1], Hyunki Hong [2,*] and Changkyoung Eem [2]

[1] Department of Integrative Engineering, Chung-Ang University, Heukseok-ro 84, Dongjak-ku, Seoul 06973, Korea; leesy88@cau.ac.kr

[2] College of Software, Chung-Ang University, Heukseok-ro 84, Dongjak-ku, Seoul 06973, Korea; ckeem@cau.ac.kr

* Correspondence: honghk@cau.ac.kr; Tel.: +82-2-820-5417

Received: 5 November 2020; Accepted: 9 December 2020; Published: 11 December 2020

Abstract: Deep learning has been utilized in end-to-end camera pose estimation. To improve the performance, we introduce a camera pose estimation method based on a 2D-3D matching scheme with two convolutional neural networks (CNNs). The scene is divided into voxels, whose size and number are computed according to the scene volume and the number of 3D points. We extract inlier points from the 3D point set in a voxel using random sample consensus (RANSAC)-based plane fitting to obtain a set of interest points consisting of a major plane. These points are subsequently reprojected onto the image using the ground truth camera pose, following which a polygonal region is identified in each voxel using the convex hull. We designed a training dataset for 2D–3D matching, consisting of inlier 3D points, correspondence across image pairs, and the voxel regions in the image. We trained the hierarchical learning structure with two CNNs on the dataset architecture to detect the voxel regions and obtain the location/description of the interest points. Following successful 2D–3D matching, the camera pose was estimated using *n*-point pose solver in RANSAC. The experiment results show that our method can estimate the camera pose more precisely than previous end-to-end estimators.

Keywords: camera pose estimation; deep learning; convolutional neural network; voxels; interest points detection and description

1. Introduction

The ability to estimate the pose of a six degree of freedom (6-DoF) camera is important in augmented reality, autonomous navigation, and robotics applications [1–5]. Such visual localization has been demonstrated with structure from motion (SfM) and simultaneous localization and mapping (SLAM) techniques [1,2,6,7], methods in which the 3D points in a scene are associated with the 2D images of these points captured with local descriptors. With these techniques, matches between 2D points in the image and 3D points (2D–3D matching) are found by searching through the descriptor space. However, the search through the shared descriptor space becomes slower and more prone to errors as the scene grows, as ambiguous matches become more likely. Specifically, the indirect SLAM estimates 3D geometry from a set of keypoint matches, optimizing a geometric error. The direct SLAM system, such as direct sparse odometry, does not need a pre-processing step like establishing correspondences and optimizes a photometric error defined directly on the images for inverse depth estimation [8]. However, the direct approach is much affected by strong geometric noise, e.g., originating from a rolling shutter or inaccurate intrinsic calibration. The reason is that the points, which have sufficiently high image gradient magnitude, of the keyframe are tracked in subsequent frames using a discrete search along the epipolar line, minimizing the photometric error.

Additionally, structure-based methods generally employ scale-invariant feature transform (SIFT) [9] as a feature descriptor, which is much affected by challenging changes in the scene, such as illumination, weather conditions, and viewpoints. Binary robust independent element features (BRIEF) select a random pair of pixels in a neighborhood. The pixels are randomly drawn from a Gaussian distribution centered around the keypoint and they are binary tested [10]. BRIEF take all the keypoints found by the FAST algorithm and convert them into a binary feature vector so that together, they can represent an object. Its performance is similar to SIFT in many respects, including robustness to lighting, blur, and perspective distortion. However, it is very sensitive to in-plane rotation. Oriented FAST and rotated BRIEF (ORB) builds on the FAST keypoint detector with orientation components and the steered BRIEF descriptor [11]. The ORB feature descriptor does not have scale invariance.

Convolutional neural networks (CNNs) have recently been used for deep learning-based visual localization and camera pose estimation [3,4,12–17], with PoseNet being the first application of CNN to end-to-end camera localization [3]. This network can estimate a 6-DoF camera's pose from an image directly, without establishing frame-to-frame feature correspondence or storing keyframes over an image sequence. With PoseNet, a deep CNN architecture generates visual features from an image, which are mapped to a localization feature vector using a fully connected layer. The final two connected layers correlate the translation and orientation information. Additionally, using transfer learning, the network can be trained from datasets with limited sizes. However, its localization error has an order of magnitude that is larger than those of comparable structure-based methods [5,6,18]. VlocNet++'s architecture [4] is based on multitasked learning for 6-DoF visual localization, semantic segmentation, and odometry estimation, exploiting the interdependencies within these tasks for more efficient operation. Because the geometric and structural information encoded by the odometry model are shared with the localization head, VlocNet++ can be trained on a dataset consisting of consecutive monocular image pairs.

Although the method above was designed to aggregate motion-specific temporal information and fuse semantic features into the localization stream based on region activations, CNNs have also been employed to estimate the pose of a known object that is rigid with respect to the camera, given an image of the object [12,13]. The main advantage of such methods is their ability to handle the occlusion problem by identifying parts of an object in cluttered scenes. The target of these studies is mainly the robotic manipulation of specific objects, such as chairs and boxes, and human–robot interaction. Deep neural networks have also been used for 6-DoF tracking of rigid objects in large occluded environments [16,19]. However, because these methods require red, green, blue, and depth (RGB-D) sensing data, their practical use is constrained to indoor scenes.

Although camera pose estimation has been more precise with recent structure-based methods than with learning-based models, their performance is affected by the limitations of the feature tracking techniques. For instance, SIFT-based SfM methods require that the temporal interval between the two images is very short for feature tracking to work successfully [9]. In the case of a small baseline, most regions of the scene overlap in consecutive images. However, feature tracking fails if the baseline is too high, as the regions in consecutive images no longer overlap. Since deep learning-based models can learn scene features, they estimate the camera pose directly from a single image, with no need for feature tracking. Thus, many end-to-end deep learning methods have near real-time inference times and an appealing simple pipeline. In contrast, SfM methods need large-scaled prior information about the scene to estimate the camera pose from an input image. This means that SfM methods need higher computational and memory costs than deep learning models. However, the localization errors obtained by deep learning models have generally been larger than with structure-based methods.

Coarse-to-fine localization using learned image-wide global (for image retrieval) and local descriptors (for 2D–3D matching) has been introduced to improve the localization performance of deep learning-based methods [14,15]. Here, the retrieval process is used to obtain the k-closest images to a given image, for localization at the map level (location hypotheses). A precise pose is subsequently estimated based on 2D–3D matches within this candidate space. While SIFT was used as a local descriptor in [14], with subsequent matches made using a k-d tree, it generates a large number of features, making 2D–3D matching particularly computationally expensive, with real-time local matching based on SIFT being similarly expensive. To address this, the hierarchical feature network (HF-Net) detects interest points and computes local and global descriptors collectively, maximizing the sharing of computations [15]. This model detects and matches interest points, which are reconstructed into sparse 3D models with SfM techniques, using SuperPoint [20]. However, studies of hierarchical models, such as those detailed above, do not present any scheme for constructing a training dataset for 2D–3D matching, and in such models, the limitations of the SfM technique remain unsolved.

Many deep learning methods have been proposed for establishing a pixel-level correspondence between images [20–23]. The approach adopted by the SuperPoint architecture is to reduce the dimensionality of the image to be processed with a Visual Geometry Group at University of Oxford (VGG)-style [24] encoder. Subsequently, regression tasks are divided by splitting the architecture into two decoder heads, for pixel-level interest point detection and for interest point description [20]. SuperPoint requires point pre-training from a synthetic dataset consisting of simple geometric shapes with no ambiguity in the interest point locations. Additionally, to help the point detector recognize the scene from several different viewpoints and scales, homographic adaptation is applied to warp the input image. Since changes in homography do not sufficiently reflect possible visual changes in scenes that have several complicated surfaces, Christopher et al. included a correspondence contrastive loss function in a fully convolutional architecture, and a convolutional spatial transformer to mimic SIFT's patch normalization [22]. Similarly, with D2-Net, detection and description parameters are shared, with a joint formulation used for both tasks [21]. Hence, to train deep learning models to identify detected interest points based on a description, the training dataset should include a large number of image pairs and correspondence point sets, as well as consider effects such as illumination and viewpoint changes.

The goal of our study is to estimate a 6-DoF camera's pose from an input image related to a learned representation of a known scene. To train the deep learning-based camera pose estimator and evaluate its performance, we require image sets consisting of scenes and the ground truth poses of the cameras that captured them. Since the camera pose is related to the coordinates of a 3D model, a dataset (typically a 3D point cloud) that considers this, such as Kendall's outdoor urban localization dataset, Cambridge Landmarks [3], is also necessary. These datasets included input images and 3D points reconstructed using the SfM technique [2] and correspondence sets over images to be used as training and test data. This dataset can be downloaded at the following URL (http://mi.eng.cam.ac.uk/projects/relocalisation/). The log file informs us which image sequence is the training or test dataset. Our training dataset consists of 3D points and image pairs with point-based correspondence. Here, although some image pairs are from consecutive frames, we have also included image pairs with a large baseline, to ensure the effects of changes in camera viewpoint and illumination can be considered in our training procedure.

In this study, scenes were digitized using identical voxels, consisting of 3D points in the scene space, divided according to their global location. The main plane, i.e., the plane with the most 3D points in each voxel (typically an exterior plane like a wall or the ground in the scene), is extracted using random sample consensus (RANSAC)-based plane fitting. Then, two networks are trained; the segmentation model, to learn the features of inlier 3D points in the voxel and then, detect the voxel regions in a query image, and the SuperPoint model [20], to extract the location and description of the interest points from the image, which correspond to the inlier 3D points on the main plane. Additionally, since the interest points are distributed on the main plane of the voxel, their correspondence set across images satisfies the homography relation. Therefore, the voxel-based scene representation and training dataset architecture proposed is useful in pixel-level correspondence studies techniques on deep learning [21,22].

A flow chart of the proposed system is depicted in Figure 1. Here, 2D–3D Matching is performed by searching through the learned descriptor space. In our scene representation, each voxel is labeled with a unique identification number. We examine whether the voxel regions detected in an input image maintain geometric consistency (such as left–right order). This process enables us to remove false matches among 2D–3D correspondences. Given accurate 2D–3D matches, we employ an n-point pose (PnP) solver in the RANSAC loop to estimate 6-DoF camera poses [25]. The rest of the paper is organized as follows. In Section 2, we describe the scene representation, feature detection, and camera pose estimation processes employed in our voxel-based technique. The experimental setup used in this study is detailed in Section 3. Then, we discuss the results of our experiments in Section 4 and conclude the paper in Section 5.

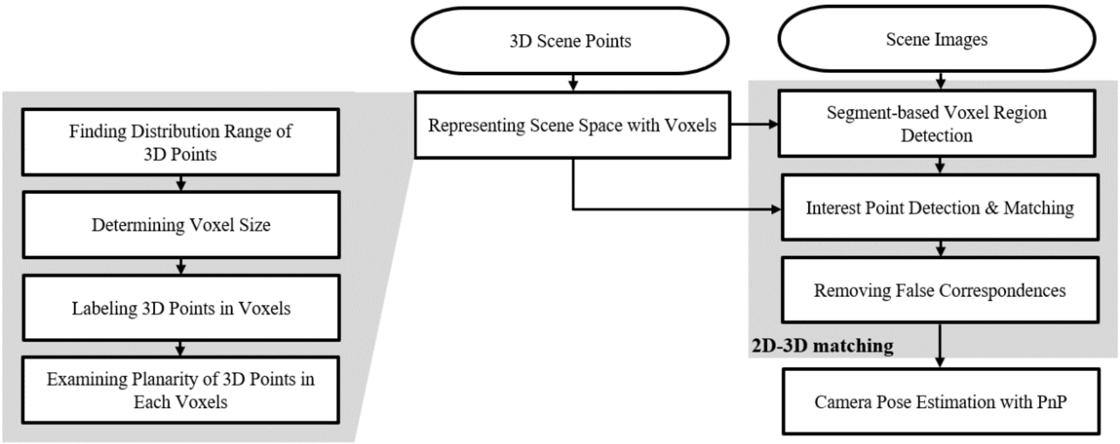

3D Scene Points
Scene Images
Representing Scene Space with Voxels
Finding Distribution Range of 3D Points
Determining Voxel Size
Labeling 3D Points in Voxels
Examining Planarity of 3D Points in Each Voxels
Segment-based Voxel Region Detection
Interest Point Detection & Matching
Removing False Correspondences
2D-3D matching
Camera Pose Estimation with PnP

Figure 1. Flow chart of the proposed method.

2. Proposed Method

In most studies, a sparse 3D model consisting of an image set of scenes and 3D points (sparse 3D model) has been used for deep learning-based camera pose estimation [12–15,19]. Specifically, 2D–3D matching has been investigated in coarse-to-fine localization studies focusing on improving the performance of deep learning methods [14,15]. By employing this technique, these methods conduct a global retrieval process to obtain a hypothesis location, and only match local features within the candidate spaces identified. The approach we describe in this paper is of a similar hierarchical nature. Here, we consider first the overall distribution of the scene elements (voxels), and then, perform local feature matching for each voxel region. In addition, for 2D–3D matching, we trained the two separate models for coarse and fine localization on datasets consisting of 3D points and the set of corresponding points in the image pairs, as detailed above.

2.1. *Voxel-Based Scene Representation*

To build a training dataset suitable for 2D–3D matching, we represent the scene as a regular tessellation of cubes in Euclidean 3D space. Each cube, or voxel, has six equal square sides, and is uniquely indexed with an identification number based on its global location. Outliers, such as noisy 3D points, that are very distant from the cloud distribution of 3D points in the scene are removed first. Then, the number and the size of voxels are determined based on the scene volume and the number of 3D points.

A summary of the numbers of images and 3D points in four scenes (King's College, Old Hospital, Shop Façade, and St. Mary's Church) from the Cambridge Landmarks dataset [3] is shown in Table 1. The number of training/test images of King's College, Old Hospital, Shop Façade, and St. Mary's Church is 1220/343, 895/182, 231/103, and 1487/530, respectively. Here, the number of voxels details how finely the scene space is divided. Each cube, including the scene space, is segmented as a 3D grid. For example, if the number of voxels is 50, the scene space is composed of 125,000 voxels (50 × 50 × 50 voxels). Since, in these datasets, 3D points were originally reconstructed using the SfM technique, the number of 3D points is dependent on the number of images and the spatial extent of the scenes in meters.

Table 1. Summary of the key scene representation metrics for the scenes from the Cambridge Landmarks dataset [3], according to the number of voxels.

	King's College			**Old Hospital**			**Shop Façade**			**St Mary's Church**	
Number of voxels	50	100	120	15	20	30	20	30	40	50	100
Avg. number of inlier 3D points per voxel	426	323	271	295	195	139	299	197	139	995	403
Avg. distance errors (cm)	3.3	3.4	3.5	3.4	3.4	3.4	3.1	3.1	3.0	3.4	3.3
Number of inlier voxels	86	140	194	120	229	318	95	157	241	150	474
Total number of images	1563			1077			334			2017	
Spatial Extent (m)	140 × 40			50 × 40			35 × 25			80 × 60	
Number of 3D Points	171,904			109,381			59,490			419,837	

In general, most 3D points in a scene (reconstructed by SfM or captured by RGB-D sensors such as Kinect) consist of the exterior walls of a building, the ground, and the outer faces of an object. Depending on the size of a voxel, the set of 3D points in the regular grid may be distributed on a main plane and/or on several small-sized faces. Points that are distributed on a main plane are more likely to be captured from multiple viewpoints by the camera, since points in other locations tend to be occluded due to the geometric complexity of the projecting face. To build a training dataset, we determine interest points that are located on the main plane of each voxel. In previous studies of end-to-end estimators, it has been suggested that based on the analysis of a saliency map, textureless surfaces such as road and sky may prove informative [3]. However, because there is no 3D information in these regions, they are unsuitable for 2D–3D matching.

Inlier 3D points located on the main plane of a voxel were determined using RANSAC-based plane fitting. With this approach, a hypothesized plane is fit to a set of 3D points, and points are only accepted as inliers if their distances to the plane are below a predefined threshold. To maximize the total number of inlier 3D points for a hypothesized plane, this process is repeated. An illustration of the distribution of 3D points in our voxel representation, as well as inlier 3D points identified following RANSAC-based plane fitting, is shown in Figure 2. Here, each voxel is labeled uniquely with an identification number, while 3D points in each voxel are marked with different colors. In this study, interest points for our training dataset architecture are defined as inlier 3D points and their corresponding 2D positions.

The average number of inlier 3D points for each voxel, as well as their average distance from the plane, are also summarized in Table 1. From this, it can be seen that in spite of the number of voxels in the scene space, there are very few voxels with inlier 3D points. For example, in the King's College scene, only 86 voxels of a possible 125,000 were identified as containing inlier 3D points. This constitutes a challenge for localization; fewer inlier 3D points are found in each voxel if the scene space is digitized more finely, making it difficult to extract voxel regions from the image. Conversely, the areas inlier voxels occupy in an image decrease (i.e., they are more precisely located) as the number of inlier voxels increases. Since the ground truth camera poses are included in the training dataset, the coordinates (x and y) of the inlier 3D points in the image can be computed using geometrical reprojection. By applying the convex hull operation to the 2D point set corresponding to the inlier 3D points, we can obtain an image region for each inlier voxel. As the convex hull of the inlier set is the smallest convex polygon that contains all the inlier points, the apparent contradiction of requiring multiple voxels in a small region can be addressed. In the training data structure, inlier image regions are labeled with the identification number of the voxel.

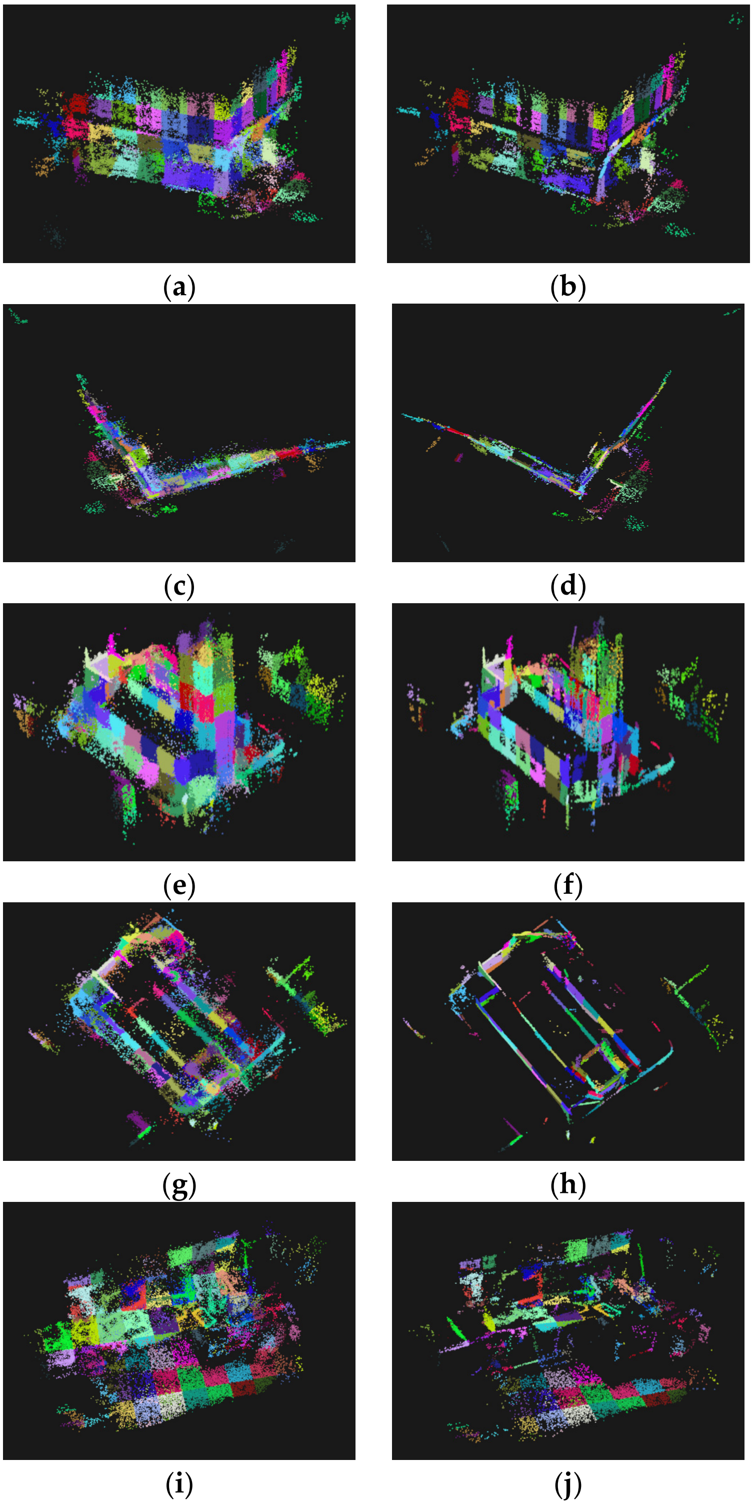


Figure 2. (**a**) Voxel-based representation of the Shop Façade scene. (**b**) Inlier 3D point set for the Shop Façade scene after random sample consensus (RANSAC)-based plane fitting. (**c**,**d**) Top views of (**a**,**b**), respectively. (**e**) Voxel-based representation of St Mary's Church scene. (**f**) Inlier 3D point set for St. Mary's Church scene. (**g**,**h**) Top views of (**e**,**f**), respectively. (**i**) Voxel-based representation of the Heads scene. (**j**) Inlier 3D point set for the Heads scene.

In this study, labeled regions in an input image were detected using AdapNet, originally proposed for semantic segmentation with complementary modalities and spectra, such as RGB and depth [26,27]. Here, AdapNet was trained on the indexed images with the voxel's identification number. In other words, the residual learning model for semantic segmentation is trained to detect the image regions labeled with the voxel's identification number. As AdapNet is based on ResNet, it includes batch normalization and skip convolution layers. In our experiments, we changed the dimensions of the convolutional filter before the skip connection, modifying two feature maps of the size $12 \times 48 \times 96$ to two feature maps of the size $64 \times 48 \times 96$. As CNNs require large amounts of training data, the ResNet50 model was initially pretrained using the ImageNet [28]. Following this, AdapNet learns landmark features in each voxel region, their spatial distribution, and the alignment of voxel regions in our training dataset.

To minimize false detection of the voxel regions, these are examined based on their spatial relation to ensure that the geometric consistency of the scene representation, such as left–right order, is satisfied. Here, we assume that no camera rolling (rotation in the z-axis direction) occurs. Voxels in the scene are assigned a unique identification number sequentially, according to their location in the initial scene representation setup. This enables us to reference the original scene when examining the geometric relationship between voxels, such that falsely detected regions can be deleted. Voxel regions with areas that are too small are also removed.

2.2. Interest Point Detection and Description

In this study, we defined our training dataset to include the inlier 3D points and the inlier voxel regions. To train the detector and descriptor network to identify interest points (inlier 3D points and their 2D reprojections), correspondence between image pairs is also required. Hence, in our architecture, the image pairs to be included in the training dataset are determined based on two conditions. First, we define a threshold for the minimum number of points of correspondence required for training (set to 80 in experiments). Then, we identify image pairs with the minimum number of interest points and define a fraction (set to 40% in experiments) to be used for the selection of image pairs. To be included in the training dataset, the ratio of the number of points of correspondence to the number of interest points should be above this fraction. With this procedure, our training dataset can include image pairs with a large baseline, unlike with previous techniques, which were restricted to using pairs of consecutive monocular images [4]. Hence, with this training architecture, the deep learning models can consider the effects of changes in camera viewpoints and illumination, as illustrated in Figure 3. Furthermore, the size of the training image set is also increased, because the number of combinations of image pairs with sufficient correspondence is much larger than the number of test images.

Interest points in the input image are detected as reprojections of the inlier 3D points from extracted voxel regions. The description information for these points is also required to complete the matching procedure. Here, we employ the SuperPoint model as the detector and descriptor network for the interest points, as it produces pixel-level interest point locations accompanied by L2-normalized descriptors. To address the possibility of missing interest points, SuperPoint applies homographic adaptations during training [20], for which it requires interest points before training, as homography is a good model for the transformation that occurs when the same 3D point is seen from different viewpoints. This homographic adaptation is most effective with planar scenes or scenes, such as panoramas, where most elements are far away from the camera. However, in general, there are several different planes in outdoor scenes, and the location of the camera when capturing the target scene cannot always be dictated. With our training architecture, interest points and their 2D correspondence set are provided in image pairs. Since these interest points are distributed on the main plane of a voxel, geometric variations across images can be considered by homographic transformation. Hence, the computationally intensive pre-training steps in the original SuperPoint model can be avoided, since the interest points in the voxel in one image and their correspondence points in another image have homographic relation.

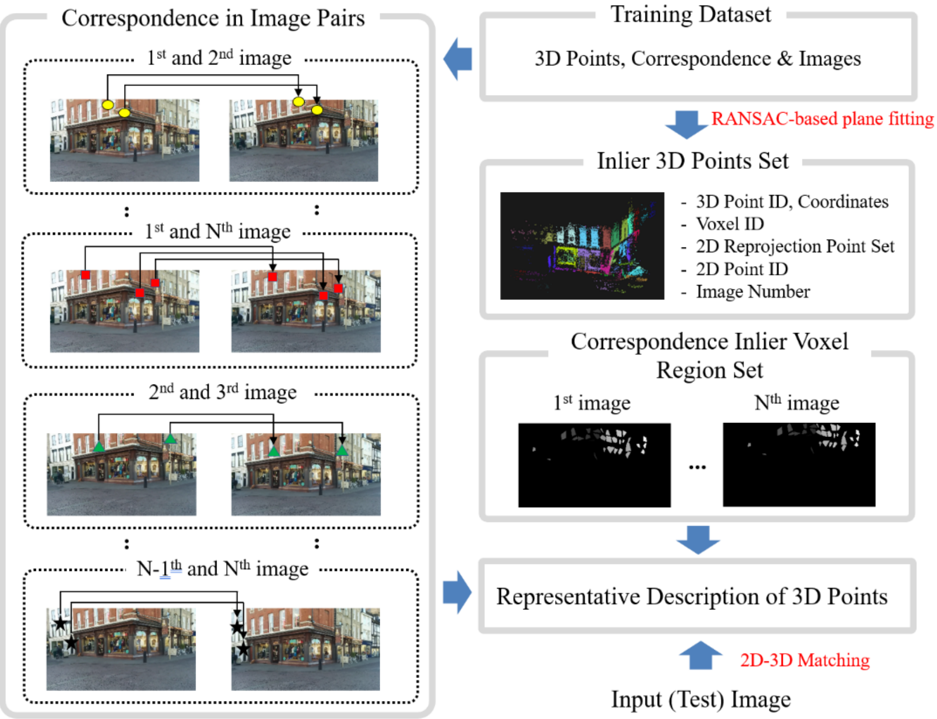

Correspondence in Image Pairs
1st and 2nd image
1st and Nth image
2nd and 3rd image
N-1th and Nth image
Training Dataset
3D Points, Correspondence & Images
RANSAC-based plane fitting
Inlier 3D Points Set
- 3D Point ID, Coordinates
- Voxel ID
- 2D Reprojection Point Set
- 2D Point ID
- Image Number
Correspondence Inlier Voxel Region Set
1st image
Nth image
Representative Description of 3D Points
2D-3D Matching
Input (Test) Image

Figure 3. Overview of the training dataset architecture.

An illustration of the two networks used for voxel region detection and interest point detection/description is shown in Figure 4. Here, interest points are identified using the voxel regions detected in the input image. To decrease the number of candidate interest points used in descriptor matching, we examine their positions first, and then, compare their description information, ensuring that only interest points in the detected voxel are evaluated in this process. In this study, we obtained description vectors for 3D inlier points across multiple images, and used the average of these vectors as representative description information. Here, the descriptor used by SuperPoint has 256 dimensions. To complete the localization, the description information of an interest point in a query image is compared with the representative description of 3D points, with the nearest neighbor search used as the metric for point matching. The voxel's identification number and the positions (X, Y, and Z coordinates) of 3D points can be accessed from the matched interest points in the data structure for 2D–3D matching.

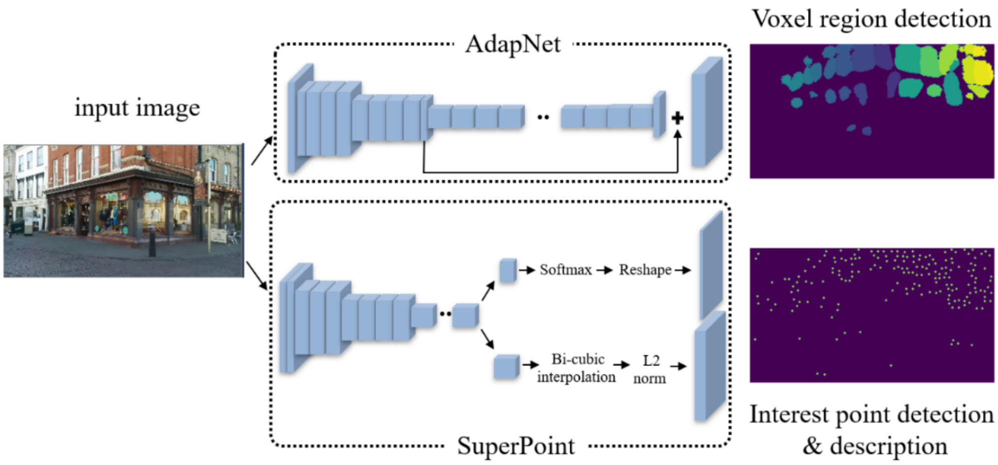


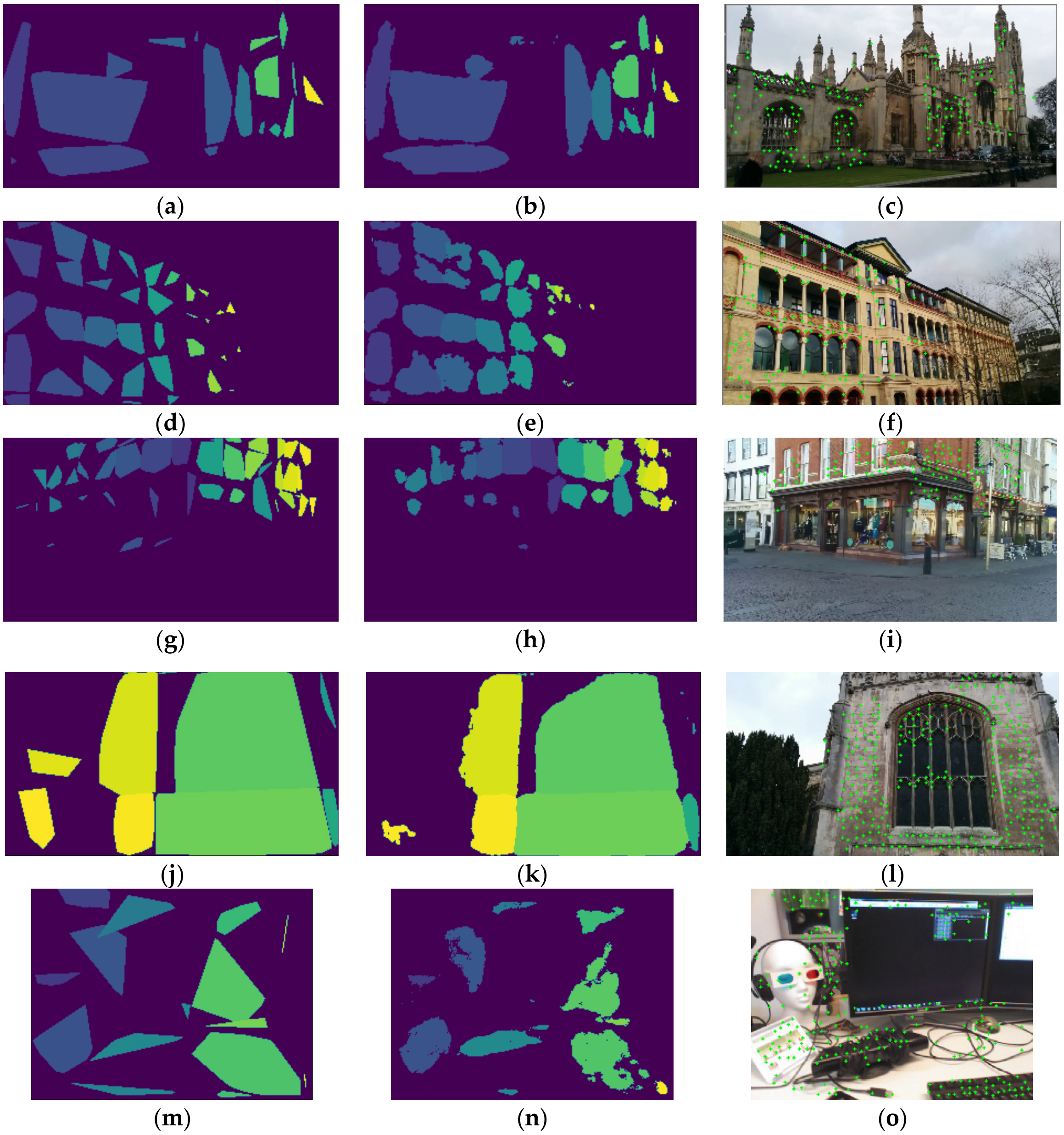
Figure 4. Illustration of the deep learning network architecture used in this study. Separate convolutional neural networks were used for voxel region detection and interest point detection/description.

Once matches have been made between 2D and 3D data, a PnP solver included inside the RANSAC loop estimates the camera pose. Here, in addition to the labeled regions, we consider the center of the labeled regions as a potential point of interest, if the number of interest points detected in the voxel regions is less than a given threshold (typically, four points are necessary for the PnP algorithm). Our use of the convex hull algorithm ensures that labeled regions are polygonal. As such, extrapolation errors are somewhat avoided with this procedure, since the centroids of the labeled regions are within these polygons. The 3D position of the centroid is determined as the closest 3D point to the centroid obtained by averaging the inlier 3D points in the voxel.

3. Experimental Setup

To test our pose estimation method, we conducted experiments on a computer equipped with a 2.9 GHz central processing unit (CPU) and an NVIDIA GTX 2080Ti graphics processing unit (GPU), using TensorFlow and the PyToch deep learning library. Here, we used the four scenes from the outdoor Cambridge Landmarks library mentioned in Section 2 to evaluate the performance of the camera pose estimator. These datasets contain images of large scale outdoor urban environments captured from distant walking paths. As a result, there is a significant amount of urban clutter, such as pedestrians and vehicles, in different scenes. Images were collected at several different points in time, giving a variety of lighting and weather conditions. We selected a subset of images from the four different scenes for use as the training dataset, retaining the rest for use as the test dataset. To examine the effects of the number of voxels on estimation performance, we conducted two different sets of experiments, termed Case 1 and Case 2. In Case 1, the King's College, Old Hospital, Shop Façade, and St Mary's Church scenes were represented with 50, 15, 20, and 50 voxels, while in Case 2, the same four scenes were represented with 100, 30, 40, and 100 voxels, respectively. Additionally, we used the Heads scene among seven scenes, which contain significant variation in camera height and are designed for RBG-D relocalization [29]. The Heads scene, which consisted of 1000 training datasets and 1000 test datasets, was represented with 15 voxels (Case 1).

We employed a rectified linear unit as the activation function for the neural network, while the adaptive moment estimation (Adam) optimizer was used to update its weight and bias parameters. The learning rate for the latter was set to 0.0001, while the exponential decay rates for the moment estimates were set to 0.9 and 0.999. Mini-batches of sixteen and four images were defined for AdapNet and SuperPoint, respectively. Here, input images for AdapNet and SuperPoint were resized to 304 × 168 pixels. For region detection, we configured AdapNet's cost function to minimize cross-entropy (softmax) loss, while for localization, SuperPoint was set up to optimize three losses—two for the interest point detectors across an image pair and one for the descriptor, simultaneously. Here, to consider both negative and positive correspondences, we set the weighting term to 25. A separate weighting term was set to 100 to balance detection and descriptor losses. To identify voxel regions in the input image, candidate voxels selected by AdapNet were evaluated using the mean IOU (intersection over union) index, also known as the Jaccard index, a commonly used metric for comparing the area similarity of two arbitrary shapes [30]. To identify camera poses, we use OpenCV's implementation of a pose estimation algorithm (solvePnP) inside the RANSAC loop [31]. Given a set of pose hypotheses, each candidate pose is evaluated with respect to the 2D–3D correspondence points and assigned a score based on its inlier count. The pose with the highest count is selected as the final estimate. Figure 5 shows the image regions of the inlier 3D points in our voxel-based scene representation, the detected voxel regions, and the interest points in the detected voxel regions for Case 1.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**) (**i**) (**j**) (**k**) (**l**) (**m**) (**n**) (**o**)

Figure 5. Illustration of the 2D–3D matching process with our voxel-based scene representation system: Voxel regions of inlier 3D points in (**a**) King's College, (**d**) Old Hospital (**g**) Shop Façade, (**j**) St. Mary's Church scenes, and (**m**) Heads; Voxel regions detected in (**b**) King's College, (**e**) Old Hospital (**h**) Shop Façade, (**k**) St. Mary's Church images, and (**n**) Heads; Interest points in detected voxel regions in (**c**) King's College, (**f**) Old Hospital (**i**) Shop Façade, (**l**) St. Mary's Church images, and (**o**) Heads.

4. Results and Discussion

A summary of the results of pose estimation conducted using our framework is detailed in Table 2. Here, the framework is evaluated based on the precision of point detection and region identification. For the former, potential 3D points corresponding to the detected interest points are reprojected onto the image. The distance between the detected point and the reprojected point is calculated, and the former is accepted as being accurate if this distance is less than 5 pixels. This limits the size of the point detection error in Table 2, since the maximum distance between potential interest points and detected interest points is 5 pixels. By using a low point-ness threshold, the number of points to be evaluated for detection can be increased as much as possible, regardless of their precision.

Table 2. Comparison of point detection performance by scene and experimental configuration.

		King's College	Old Hospital	Shop Façade	St Mary's Church
Case 1	Average IOU	0.94	0.91	0.90	0.90
	Point Detection Precision	0.65	0.47	0.55	0.51
Case 2	Average IOU	0.93	0.92	0.92	0.89
	Point Detection Precision	0.71	0.59	0.65	0.59

Similar region detection results were noted for Case 1 and Case 2; with both of these cases, the average of IOU values from all four scenes was 0.92. This result indicates that the detection of voxel regions becomes trivial above a certain size, as these have enough visual features for them to be learned by the neural network. Hence, beyond this threshold, the division of the scene space does not affect region detection. In contrast, it can be noted that interest point detection is better with Case 2 than with Case 1, as there are more inlier 3D points in the former dataset than in the latter. For example, from Table 1, it can be seen that there are 86 inlier voxels in the King's College scene when it is defined as in Case 1, and on average, there are 426 inlier 3D points per voxel. Therefore, in this case, 36,636 inlier 3D points are considered in the training dataset. Conversely, for the same scene, in Case 2, there are 194 inlier voxels and an average of 274 inlier 3D points per voxel. Thus, for this training dataset, there are 52,574 inlier 3D points. This increase in the number of correspondence sets used in SuperPoint's training explains the improved performance noted in Case 2. Similarly, this increased amount of training data improves the technique's camera pose estimation. Table 3 details a comparison of camera pose estimation performances using the two configurations of our technique, and previously developed estimator models, in terms of median translation and rotation errors. The experimental parameters in previous methods have a different setup. For example, PoseNet rescaled the input image so that the smallest dimension was 256 pixels and randomly cropped to the 224 × 224 pixels in the training phase. This model was evaluated on a single 224 × 224 center crop [3]. In Table 3, the median values of the obtained camera pose results from the test dataset are presented. The localization errors obtained with end-to-end deep learning methods have been larger than with structure-based methods. Our method achieves comparable results to the SfM method to some extent and has near real-time inference times and an appealing simple pipeline. Here, it can be seen that the performance for Case 2 is better than for Case 1, for all four scenes.

Table 3. Comparison of translation and rotation errors observed with different camera pose estimation methods.

		King's College	Old Hospital	Shop Façade	St Mary's Church	Heads
PoseNet [3]	Rotation (°)	5.40	5.38	8.08	8.48	12.0
	Translation (m)	1.92	2.31	1.46	2.65	0.29
LSTM-Pose [32]	Rotation (°)	3.65	4.29	7.44	6.68	13.7
	Translation (m)	0.99	1.51	1.18	1.52	0.21
PoseNet2 [33]	Rotation (°)	1.04	3.29	3.78	3.32	13.0
	Translation (m)	0.88	3.20	0.88	1.57	0.17
VLocNet [34]	Rotation (°)	1.42	2.41	3.53	3.91	6.64
	Translation (m)	0.84	1.08	0.59	0.63	0.05
Active Search [6]	Rotation (°)	0.6	1.0	0.4	0.5	1.5
	Translation (m)	0.42	0.44	0.12	0.19	0.02
AdapNet+SuperPoint (Case 1)	Rotation (°)	1.42	3.39	1.68	1.69	2.86
	Translation (m)	0.62	0.96	0.20	0.49	0.74
AdapNet+SuperPoint (Case 2)	Rotation (°)	1.26	2.85	2.00	1.62	-
	Translation (m)	0.61	0.77	0.19	0.50	-

A comparison of the duration of each computation step by scene and experimental case is included in Table 4. The results above have shown that with our voxel-based scene representation, the number of voxels affects computation performance. This is also reflected in Table 4. Representing the scene with more voxels decreases the average area of a voxel region. As such, fewer 3D points are included in a voxel region, and both the number of interest points and the size of the representative description information vector are decreased. As a result, the number of candidate pairs considered in the description matching procedure is reduced, such that the duration of this process, as well as the pose estimation procedures, is reduced, as shown in Table 4. Conversely, since the number of voxels in the scene increases, the number of inlier voxels also increases. Hence, in general, the number of voxel regions to be classified in an image increase. In addition, AdapNet's weight parameters are also made larger, as this is required to classify more voxel classes. The combination of these effects means that it takes much more time to detect the voxel regions in images with Case 2 than with Case 1. Hence, since voxel region detection takes up more than half of the total computation time, in general, computation is faster with Case 1 than with Case 2. To illustrate this further, it can be noted that it takes much more time to detect the voxel regions in the Old Hospital and St Mary's Church scenes, as they have more inlier voxels than the other two scenes, as detailed in Table 1. Hence, for computational efficiency, in future studies, we will consider dilated convolution [35] to reduce the number of weight parameters the decoder in AdapNet requires. Finally, from Table 4, it can be noted that point detection, in general, takes longer with Case 2 than with Case 1. This is because the increased number of voxels typically means that more inlier 3D points and interest points are detected in Case 2 than in Case 1. As such, more interest points are considered for bi-cubic interpolation in SuperPoint. Similarly, point description and description matching take longer with the King's College and St Mary's Church scenes, since they have more inlier 3D points per voxel on average than the other two scenes, and more candidate pairs require consideration. To examine further performance of our model, we include the mean and standard deviation of the estimated camera pose results in Table 5. Some papers [15,18,21] have reported the localization rate, defined by computing the percentage of images localized within a given camera pose error threshold (for example, with translation and rotation errors smaller or equal to 0.25 m and 2 degrees). Tables 5 and 6 show that the result errors in Case 2 are more stable than those in Case 1.

Table 4. Comparison of computation time (ms) required for individual procedures in camera pose estimation.

		King's College	Old Hospital	Shop Façade	St Mary's Church
Case 1	Voxel Region Detection	27.8	31.0	27.1	35.1
	Point Detection/Description	5.4	4.4	4.0	6.5
	Description Matching	18.1	5.2	4.9	16.1
	PnP	0.7	0.9	0.7	0.9
	Total Computation Time	52.0	41.5	36.7	58.6
Case 2	Voxel Region Detection	31.3	64.2	55.2	85.0
	Point Detection/Description	6.2	4.6	4.2	8.0
	Description Matching	11.9	4.9	4.1	12.9
	PnP	0.6	0.7	0.6	0.9
	Total Computation Time	50.0	74.4	64.1	106.8

Table 5. Comparison of mean and standard deviation (std. dev.) of translation and rotation errors.

		King's College		**Old Hospital**		**Shop Façade**		**St Mary's Church**		**Heads**
		Case 1	**Case 2**	**Case 1**	**Case 2**	**Case 1**	**Case 2**	**Case 1**	**Case 2**	**Case 1**
Rotation	mean	2.44	1.63	9.50	7.83	3.27	3.49	3.06	2.34	8.34
(°)	Std. dev.	3.45	1.76	15.90	14.56	4.22	5.25	10.58	5.00	21.46
Translation	mean	0.97	0.76	2.59	1.89	0.36	0.36	1.22	0.82	1.90
(m)	Std. dev.	3.54	0.71	4.50	3.65	0.41	0.62	6.19	1.65	4.64

Table 6. Evaluation of the localization rate within a given camera pose error threshold.

		King's College		**Old Hospital**		**Shop Façade**		**St Mary's Church**		**Heads**
		Case 1	**Case 2**	**Case 1**	**Case 2**	**Case 1**	**Case 2**	**Case 1**	**Case 2**	**Case 1**
m/degree	0.25/2.0	0.19	0.20	0.13	0.08	0.58	0.65	0.18	0.16	0.10
	0.5/5.0	0.39	0.43	0.31	0.35	0.79	0.80	0.49	0.49	0.32
	5.0/10.0	0.98	0.99	0.80	0.84	0.92	0.94	0.97	0.98	0.87

5. Conclusions

In this paper, we discussed a method for improving the performance of end-to-end camera pose estimation. First, we represent the scene space as a series of voxels, defined according to the volume of the scene and the number of 3D points. Then, we determine the set of inlier points that best represent the scene. Using RANSAC-based plane fitting, we extract interest points from 3D points located on the main plane of a voxel and apply the convex hull operation to the image reprojection of the inlier 3D points to obtain image regions for each voxel. We subsequently introduce a new training dataset architecture that includes the voxel image regions, the interest points, points of correspondence in image pairs, and the inlier 3D points. Since, based on the distribution of 3D points, the interest points obtained using our procedure satisfy homography, identifying the correspondences in image pairs makes training of the deep learning models for feature point detection and description more efficient, as variations in visual appearance are considered implicitly. Finally, we use a hierarchical structure for coarse-to-fine localization, training one deep learning model to extract voxel regions of inlier 3D points from an input image, and a separate one to obtain the interest points and their description information. Once 2D interest points and 3D inliers have been matched, a PnP inside the RANSAC loop estimates the 6-DoF camera pose. The results of our experiments show that our deep learning model can estimate camera poses more precisely than previously developed end-to-end estimation methods. In future studies, we aim to modify the AdapNet-based model to improve its computational efficiency.

Author Contributions: S.L. and H.H. proposed the idea of this paper; S.L., H.H. and C.E. reviewed this paper and provided information; S.L. and H.H. conceived and designed the experiments; S.L. and H.H. performed the experiments; S.L., H.H. and C.E. reviewed the codes in this paper and wrote this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1B07049932).

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

References

1. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 1–10.
2. Wu, C. Towards linear-time incremental structure from motion. In Proceedings of the International Conference on 3D Vision, Seattle, WA, USA, 29 June–1 July 2013; pp. 127–134.

3. Kenall, A.; Grimes, M.; Cipolla, R. PoseNet: A convolutional network for real-time 6-DOF camera relocalization. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2938–2946.
4. Radwan, N.; Valada, A.; Burgard, W. VLocNet++: Deep multitask learning for semantic visual localization and odometry. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4407–4414. [CrossRef]
5. Sattler, T.; Zhou, Q.; Pollefeys, M.; Leal-Taixe, L. Understanding the limitations of CNN-based absolute camera pose regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 3302–3312.
6. Sattler, T.; Leibe, B.; Kobbelt, L. Efficient & effective prioritized matching for large-scale image-based localization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *39*, 1744–1756. [PubMed]
7. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera slam. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef] [PubMed]
8. Engel, J.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
9. Lowe, D.G. Distinctive image features from scale invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
10. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. BRIEF: Binary robust independent elementary features. In Proceedings of the 11th European Conference on Computer Vision, Heraklion, Crete, Greece, 5–11 September 2010; pp. 778–792.
11. Rublee, E.; Rabaut, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the IEEE Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
12. Xiang, Y.; Schmidt, T.; Narayanan, V.; Fox, D. PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes. In Proceedings of the Robotics: Science and Systems XIV, Pittsburgh, PA, USA, 26–30 June 2018; pp. 1–10.
13. Crivellaro, A.; Rad, M.; Verdie, Y.; Yi, K.M.; Fua, P.; Lepetit, V. Robust 3d object tracking from monocular images using stable parts. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1465–1479. [CrossRef] [PubMed]
14. Sarlin, P.; Debraine, F.; Dymczyk, M.; Siegwart, R.; Cadena, C. Leveraging deep visual descriptors for hierarchical efficient localization. In Proceedings of the 2nd Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018; pp. 456–465.
15. Sarlin, P.; Cadena, C.; Siegwart, R.; Dymczyk, M. From coarse to fine: Robust hierarchical localization at large scale. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 12716–12725.
16. Garon, M.; Lalonde, J. Deep 6-dof tracking. *IEEE Trans. Vis. Comput. Graph.* **2017**, *23*, 2410–2418. [CrossRef]
17. Brahmbhatt, S.; Gu, J.; Kim, K.; Hays, J.; Kautz, J. Geometry-aware learning of maps for camera localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2616–2625.
18. Shavit, Y.; Ferens, R. Introduction to camera pose estimation with deep learning. *arXiv* **2019**, arXiv:1907.05272.
19. Su, J.; Cheng, S.; Chang, C.; Chen, J. Model-based 3D pose estimation of a single rgb image using a deep viewpoint classification neural network. *Appl. Sci.* **2019**, *9*, 2478. [CrossRef]
20. DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperPoint: Self-supervised interest point detection and description. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–23 June 2018; pp. 224–236.
21. Dusmanu, M.; Rocco, I.; Pajdla, T.; Pollefeys, M.; Sivic, J.; Torii, A.; Sattler, T. D2-Net: A trainable CNN for joint description and detection of local features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 8092–8101.
22. Choy, C.B.; Gwak, J.Y.; Savarese, S.; Chandraker, M. Universal correspondence network. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 4–9 December 2016; pp. 2414–2422.
23. Altwaijry, H.; Veit, A.; Belongie, S. Learning to detect and match keypoints with deep architectures. In Proceedings of the British Machine Vision Conference, York, UK, 19–22 September 2016.
24. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.

25. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
26. Valada, A.; Vertens, J.; Dhall, A.; Burgard, W. AdapNet: Adaptive semantic segmentation in adverse environmental conditions. In Proceedings of the IEEE International Conference on Robotics and Automation, Marina Bay Sands, Singapore, 29 May–3 June 2017; pp. 4644–4651.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; Li, F. ImageNet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
29. Shotton, J.; Glocker, B.; Zach, C.; Izadi, S.; Criminisi, A.; Fitzgibbon, A. Scene coordinate regression forests for camera relocalization in RGB-D images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2930–2937.
30. Rezatofighi, H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized intersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 658–666.
31. OpenCV: Camera Calibration and 3D Reconstruction. Available online: https://docs.opencv.org/3.4/d9/d0c/group__calib3d.html (accessed on 1 October 2020).
32. Walch, F.; Hazirbas, C.; Leal-Taixé, L.; Sattler, T.; Hilsenbeck, S.; Cremers, D. Image-based localization using lstms for structured feature correlation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 627–637.
33. Kendall, A.; Cipolla, R. Geometric loss functions for camera pose regression with deep learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5974–5983.
34. Valada, A.; Radwan, N.; Burgard, W. Deep auxiliary learning for visual localization and odometry. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, Australia, 20–25 May 2018; pp. 6939–6946.
35. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolution. In Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016; pp. 1–13.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.