GEOMETRIC CAMERA POSE REFINEMENT WITH LEARNED DEPTH MAPS

Nathan Piasco [1,2], *Désiré Sidibé* [1]

[1] ViBot ERL CNRS 6000, ImViA
Université Bourgogne Franche-Comté

Cédric Demonceaux [1], *Valérie Gouet-Brunet* [2]

[2] LaSTIG, IGN, ENSG
Université Paris-Est

ABSTRACT

We present a new method for image-only camera relocalisation composed of a fast image indexing retrieval step followed by pose refinement based on ICP (Iterative Closest Point). The first step aims to find an initial pose for the query by evaluating images similarity with low dimensional global deep descriptors. Subsequently, we predict with a fully convolutional deep encoder-decoder neural network a dense depth map from the image query. We use this depth map to create a local point cloud and refine the initial query pose using an ICP algorithm.

We demonstrate the effectiveness of our new approach on various indoor scenes. Compared to learned pose regression methods, our proposal can be used on multiple scenes without the need of a specific weights-setup for each scene, while showing equivalent results.

Index Terms— Camera relocalisation, multimodal data, depth from monocular, pose estimation, ICP

1. INTRODUCTION

Image-based camera relocalisation consists in retrieving the 6 Degrees of Freedom (DoF) pose of a camera with a single image according to a known reference [1]. Initial image relocalisation methods rely on fast features indexing [2] or learned approaches [3] to quickly retrieve a precise camera pose. It is a crucial step for many applications such as robot relocalisation on a map created by a SLAM algorithm, camera tracking recovering for augmented reality or autonomous driving navigation initialisation.

In most of the cases, due to limited sensing capability of portable or embedded devices, the camera relocalisation problem has to be solved using only radiometric observation of the scene (*i.e.* images) [4]. However, recent methods of machine learning applied to computer vision are able to infer underlying geometry of a scene from single images only [5, 6]. Based on these works, we propose a new method combining both learned approaches and geometric algorithms to solve the camera relocalisation problem. We show that our method is able to quickly retrieve a pose with fast indexing of global image descriptors and then to refine the position and the orientation of the camera based on a learned representation of the scene geometry.

Our paper is presented as follows: the end of this section is dedicated to a brief review of the related work, then the details of our method are presented in section 2. The obtained results with our proposal are discussed in section 3, and we finally conclude the paper in section 4.

Related work. The state of the art on online camera localisation [2, 7] usually relies on global and local features indexing combined with costly verification steps. [7] generates virtual camera view point from a dense 3D model in order to verify the retrieved pose. In our work, the refinement step relies on the alignment of point clouds, so we do not need to construct a costly 3D dense model, neither to generate artificial data.

Recent learned approaches [8, 9] have shown impressive results in terms of accuracy and robustness to visual changes. However, these methods are scene-dependent and we need to train a whole network for each scene, narrowing the range of real applications where those methods can be used (*e.g.* real-time novel scene mapping and relocalisation [1]). Our method benefits from the robustness of learned methods, while not being specific to a particular scene since it can be deployed on multiples areas without parameters retraining.

A recent work related to ours was presented in [10]. Authors propose a two-step pose estimation algorithm where two different networks are trained, one for image retrieval by deep descriptor indexing and a second pose refinement step by relative pose estimation between two images. We use the same two-step approach but we rely on a single network and our pose refinement step involves a non-learned geometric method. Thus the learned part of our system required weakly annotated data compared to [10], and the training data are easier to gather.

2. METHOD

2.1. Method overview

Notations. We aim to recover the camera pose $\mathbf{h} \in \mathbb{R}^{4\times4}$, represented by a pose matrix in homogeneous coordinates, given an input RGB image $I \in \mathbb{R}^{3\times H\times W}$. We assume that

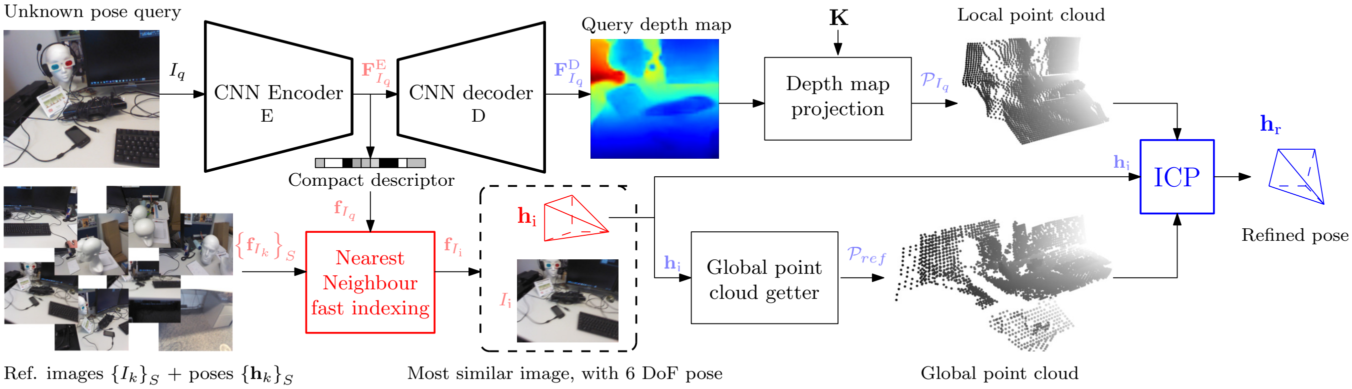
Unknown pose query
I_q
CNN Encoder E
$\mathbf{F}^{\mathrm{E}}_{I_q}$
CNN decoder D
$\mathbf{F}^{\mathrm{D}}_{I_q}$
Query depth map
K
Depth map projection
$\mathcal{P}_{I_q}$
Local point cloud
$\mathbf{h}_{\mathrm{r}}$
$\mathbf{h}_{\mathrm{i}}$
ICP
Refined pose
Compact descriptor
$\mathbf{f}_{I_q}$
$\{\mathbf{f}_{I_k}\}_S$
Nearest Neighbour fast indexing
$\mathbf{f}_{I_{\mathrm{i}}}$
$\mathbf{h}_{\mathrm{i}}$
I_{i}
$\mathbf{h}_{\mathrm{i}}$
Global point cloud getter
$\mathcal{P}_{ref}$
Ref. images $\{I_k\}_S$ + poses $\{\mathbf{h}_k\}_S$
Most similar image, with 6 DoF pose
Global point cloud

Fig. 1. **Workflow of our pose estimation method.** The first indexing step retrieves the most similar image to the query from a pool of localised reference data by comparing compact deep representation extracted from our encoder. We use the pose of the first retrieved candidate as initial pose $\mathbf{h}_{\mathrm{i}}$. In a second step, we use an ICP algorithm to align a local point cloud, projected from the query depth map inferred by our deep decoder, to a reference global point cloud, producing final refined pose $\mathbf{h}_{\mathrm{r}}$.

we know the matrix $\mathbf{K} \in \mathbb{R}^{3\times3}$ of intrinsic parameters of the camera. We denote as E, respectively D, a fully convolutional neural network encoder, respectively decoder, with trainable parameters $\mathbf{w}_{\mathrm{E}}$, respectively $\mathbf{w}_{\mathrm{D}}$. E applied on I generates C_e features maps $\mathbf{F}^{\mathrm{E}}_I \in \mathbb{R}^{C_e\times H_e\times W_e}$. We denote $\mathbf{F}^{\mathrm{D}}_I \in \mathbb{R}^{1\times H_d\times W_d}$ the 1-channel output of D applied on $\mathbf{F}^{\mathrm{E}}_I$:

$$\mathbf{F}^{\mathrm{D}}_I = \mathrm{D}(\mathbf{F}^{\mathrm{E}}_I) = \mathrm{D}(\mathrm{E}(I)).$$

Encoder E reduces the spatial resolution of the input image ($W_e < W, H_e < H$), while decoder D upsamples the spatial resolution of the encoder features ($W_d > W_e, H_d > H_e$ and $W_d \leq W, H_d \leq H$).
Finally, $\mathcal{P}$ is a set of N 3D points $\mathbf{p}$: $\mathcal{P} = \{\mathbf{p}_k\}_N$.

Workflow. The complete pipeline of our method is described in figure 1. We first create a low-dimensional descriptor $\mathbf{f}_{I_q}$ from deep features $\mathbf{F}^{\mathrm{E}}_{I_q}$ computed on the query image I_q. We evaluate the similarity of this descriptor with the pool of S reference images $\{I_k\}_S$ with respective descriptors $\{\mathbf{f}_{I_k}\}_S$. We associate the pose of the most similar image as initial pose $\mathbf{h}_{\mathrm{i}}$ for the query image (see section 2.2). In a second step, the initial pose $\mathbf{h}_{\mathrm{i}}$ is refined with a geometric-based method. We use latent image description $\mathbf{F}^{\mathrm{E}}_{I_q}$ and the decoder D to generate the depth map $\mathbf{F}^{\mathrm{D}}_{I_q}$ associated to query image I_q (see section 2.3). $\mathbf{F}^{\mathrm{D}}_{I_q}$ is used to obtain the local point cloud $\mathcal{P}_{I_q}$. We then retrieve the global point cloud $\mathcal{P}_{ref}$ centred on the initial pose $\mathbf{h}_{\mathrm{i}}$. An ICP (Iterative Closest Point) registration algorithm is finally used to align $\mathcal{P}_{I_q}$ with $\mathcal{P}_{ref}$ given the initial pose $\mathbf{h}_{\mathrm{i}}$, leading to the refined query pose $\mathbf{h}_{\mathrm{r}}$ (see section 2.4).

2.2. Image indexing

We cast the pose estimation task as a content-based image retrieval problem like in [10], since the reference data are augmented with 6 DoF pose information. Recent works have shown that deep features extracted from convolutional neural network are well suited for global image description, compared to hand-crafted features [11, 12, 13]. We consider two state-of-the art global methods, Maximum of ACtivation (MAC) [14, 13] and NetVALD [11], to describe the data by low-dimensional L_2 normalized descriptors. We first compute reference descriptors $\{\mathbf{f}_{I_k}\}_S$ from the reference images (*i.e.* images with associated 6 DoF poses) using the deep features computed by E combined with a descriptor. Then we compare the query descriptor $\mathbf{f}_{I_q}$ to the pre-computed descriptors by fast nearest neighbour indexing and retrieval:

$$\mathbf{f}_{I_{\mathrm{i}}} = NN(\mathbf{f}_{I_q}, \{\mathbf{f}_{I_k}\}_S), \tag{1}$$

where NN is the nearest neighbour matching function and $\mathbf{f}_{I_{\mathrm{i}}}$ the closest retrieved candidate to the query descriptor $\mathbf{f}_{I_q}$. We consider the pose $\mathbf{h}_{\mathrm{i}}$ associated to reference image I_{i} as an approximate pose of the image I_q.

2.3. Depth map generation

Our pose refinement process is based on geometric alignment. As we consider only image data as query, we have to find a way to generate the underlying geometry of the scene corresponding to the image. Various recent deep learning generative models are able to properly reconstruct geometry associated to radiometric data, with full supervision training [5], weakly annotated data [15] or even in a self-supervised way [16]. We train an encoder/decoder jointly and in a supervised manner in order to predict the corresponding depth associated to an image with the following loss function:

$$\begin{aligned}(\mathbf{w}_{\mathrm{E}}, \mathbf{w}_{\mathrm{D}}) &= \underset{\mathbf{w}_{\mathrm{E}},\mathbf{w}_{\mathrm{D}}}{\mathrm{argmin}} \sum_k \left\|\mathbf{F}^{\mathrm{D}}_{I_k} - \mathbf{D}_k\right\|_1, &(2)\\ &= \underset{\mathbf{w}_{\mathrm{E}},\mathbf{w}_{\mathrm{D}}}{\mathrm{argmin}} \sum_k \left\|\mathrm{D}(\mathrm{E}(I_k)) - \mathbf{D}_k\right\|_1, &(3)\end{aligned}$$

where $\{I_k, \mathbf{D}_k\}_M$ are training pairs of images with corresponding depth maps.

This is the only learned part of our localisation pipeline, which only needs weakly annotated data, paired image and depth map, for training.

2.4. Point cloud alignment

Thanks to the dense geometric information provided the encoder-decoder network, we can rely on robust ICP algorithm to refine the initial pose.

Local point cloud. With the generated depth map obtained by our neural network and the intrinsic parameters of the camera, we can project the depth map on the point cloud $\mathcal{P}_I$:

$$[\mathcal{P}_I]_{l+k\times W_d} = \mathbf{p}_{l+k\times W_d} = \left[\mathbf{F}_I^{\mathrm{D}}\right]_{k,l} \cdot \mathbf{K}^{-1}[k,l,1]^T, \quad (4)$$

where $\mathbf{p} \in \mathbb{R}^3$ is a 3D point and $[\cdot]_j$ is the subscription operator at index j. $\mathcal{P}_I$ contains $W_d \times H_d$ 3D points.

Point descriptor. Refinement with ICP involves matching corresponding points between two point clouds in order to estimate a rigid transformation that minimises the distances between the paired points. Standard approaches only consider the Euclidean distance between a single point and its nearest neighbours in the reference point cloud to establish matching, making the initial alignment between the two point clouds a crucial step to obtain correct results. We can rely on point descriptors to establish strongest matches [17]. Because the point cloud to align is generated with a deep neural network, we associate to each projected point $\mathbf{p}$ a C-dimensional descriptor $\mathbf{d} \in \mathbb{R}^C$ corresponding to the deep feature computed by the encoder E at the same spatial position:

$$[\mathcal{D}_I]_{l+k\times W_d} = \mathbf{d}_{l+k\times W_d} = \left[\mathbf{G}_I^{\mathrm{E}}\right]_{k,l}, \quad (5)$$

where $\mathcal{D}_I$ is set set of point descriptors associated to point cloud $\mathcal{P}_I$ and $\mathbf{G}_I^{\mathrm{E}} \in \mathbb{R}^{C\times H_d \times W_d}$ are the feature maps from E that have the same spatial resolution that the output of D.
Descriptors $\mathcal{D}$ are obtained without any additional computation cost because latent features $\mathbf{G}_I^{\mathrm{E}}$ have already been computed by E to produce final features $\mathbf{F}_I^{\mathrm{E}}$.

Global point cloud. To align the point cloud generated from the query, we need a reference scene geometry. We sample a pool of reference data within a fixed radius centred on the initial retrieved pose $\mathbf{h}_{\mathrm{i}}$. The reference point cloud $\mathcal{P}_{ref}$ is created by aggregating all the point clouds from these data (using ground truth or generated depth maps). We also compute point descriptors $\mathcal{D}_{ref}$ corresponding to the 3D points in $\mathcal{P}_{ref}$. The reference point cloud is computed offline and stored efficiently for fast access during query time.

Pose refinement. Final refined pose $\mathbf{h}_{\mathrm{r}}$ is given by:

$$\mathbf{h}_{\mathrm{r}} = \mathtt{ICP}(\mathbf{h}_{\mathrm{i}}, \mathcal{P}_{I_q}, \mathcal{D}_{I_q}, \mathcal{P}_{ref}, \mathcal{D}_{ref}), \quad (6)$$

where ICP is the function described in algorithm 1.
The match_points function computes pairs of similar points both based on the spatial proximity and on the descriptors similarity. In other words, two pairs of point descriptors $\{\mathbf{p}_m, \mathbf{d}_m\}$ and $\{\mathbf{p}_n, \mathbf{d}_n\}$ are matched together if:

$$m, n = \underset{k,l}{\operatorname{argmin}} \left\|[\mathbf{p}_k, \mathbf{d}_k] - [\mathbf{p}_l, \mathbf{d}_l]\right\|_2, \quad (7)$$

where $[\cdot]$ is the concatenation operator. The matches are retrieved efficiently by k-d tree fast nearest neighbour search. The relative_pose function computes the relative transformation between the matched points that minimises the Euclidean difference between the two point clouds. We embed the pose computation within a RANSAC consensus, as the point cloud may contain erroneous data because it has been generated from image-only information by our encoder/decoder.

```
Data: initial pose $\mathbf{h}_{init}$, point cloud to align $\mathcal{P}$ with
      associated descriptors $\mathcal{D}$ and reference point
      cloud $\mathcal{P}_{ref}$ with assocaited descriptors $\mathcal{D}_{ref}$
Result: final pose $\mathbf{h}_{refined}$
$\mathbf{h}_{refined} \leftarrow \mathbf{h}_{init}$;
$\mathbf{h}_{relative} \leftarrow \mathbf{1}_{4\times 4}$;
while $\|\mathbf{h}_{relative} - \mathbf{I}_{4\times 4}\|_{\mathrm{F}} \geq \epsilon$ do
    $\mathcal{P}_{aligned} \leftarrow \mathbf{h}_{refined}\mathcal{P}$;
    $\mathcal{M} \leftarrow$ match_points($\mathcal{P}_{aligned}, \mathcal{P}_{ref}, \mathcal{D}, \mathcal{D}_{ref}$);
    $\mathbf{h}_{relative} \leftarrow$ relative_pose($\mathcal{M}$);
    $\mathbf{h}_{refined} \leftarrow \mathbf{h}_{relative}\mathbf{h}_{refined}$;
end
if $\|\mathcal{M}\|_2 > \epsilon_{repro}$ then
    $\mathbf{h}_{refined} \leftarrow \mathbf{h}_{init}$;
end
```

Algorithm 1: Our ICP algorithm, see text for details about functions match_points and relative_pose. Pose refinement is rejected if the mean distance between matched points, $\|\mathcal{M}\|_2$, is superior to ϵ_{repro}.

3. EXPERIMENTS

We choose an indoor camera relocalisation scenario to evaluate the pose estimation performances of the proposed method. We use 7 scenes [3] indoor dataset for both training and testing as it contains images with ground-truth depth maps acquired with a kinect, making our network easier to train.

3.1. Implementation details

Network architecture and training. We use a custom fully convolutional network architecture inspired by U-net image translation network [19] where each feature map from the encoder is given to the decoder through skip-connection. Our network has 17M parameters, that is a slightly superior to Resnet18 and inferior to Resnet50. We train the network with adam optimizer, batch size of 24 pairs {image, depth map}, with learning rate of $1e^{-4}$, divided by two every $40k$

	Vol.	Only Indexing		Scene specific (one network trained by scene)			All scenes	
Scene	(m^3)	MAC (M)	NetVLAD (V)	M + ICP w/o dc.	M + ICP	V + ICP	V + ICP	V + ICP (GT)
Chess	6	$0.31m, 14.9^\circ$	$0.29m, 13.0^\circ$	$0.28m,\ 8.6^\circ$	$0.23m,\ 5.4^\circ$	$0.22m,\ 4.9^\circ$	$0.24m,\ 5.0^\circ$	$\mathbf{0.12m},\ \mathbf{4.5}^\circ$
Fire	2.5	$0.49m, 16.7^\circ$	$0.40m, 15.5^\circ$	$0.39m, 16.5^\circ$	$0.30m, 14.1^\circ$	$0.30m, 14.1^\circ$	$0.26m,\ 9.7^\circ$	$\mathbf{0.25m},\ \mathbf{8.9}^\circ$
Heads	1	$0.28m, 20.5^\circ$	$0.20m, 16.0^\circ$	$0.18m, 14.9^\circ$	$0.19m, 14.1^\circ$	$0.17m, 12, 9^\circ$	$\mathbf{0.16m}, 10.1^\circ$	$0.18m,\ \mathbf{9.9}^\circ$
Office	7.5	$0.46m, 16.4^\circ$	$0.38m, 13.0^\circ$	$0.41m, 13.4^\circ$	$0.36m, 11.3^\circ$	$0.30m,\ 8.6^\circ$	$0.32m,\ 7.8^\circ$	$\mathbf{0.22m},\ \mathbf{7.3}^\circ$
Pumpkin	5	$0.50m, 15.0^\circ$	$0.43m, 13.1^\circ$	$0.40m, 12.0^\circ$	$0.35m,\ 7.4^\circ$	$0.34m,\ 6.8^\circ$	$0.37m,\ 6.9^\circ$	$\mathbf{0.21m},\ \mathbf{6.2}^\circ$
Red Kitchen	18	$0.30m, 11.2^\circ$	$0.23m,\ 9.5^\circ$	$0.24m,\ 7.5^\circ$	$0.19m,\ 4.9^\circ$	$0.18m,\ 4.6^\circ$	$0.23m,\ 5.0^\circ$	$\mathbf{0.15m},\ \mathbf{4.5}^\circ$
Stairs	7.5	$0.64m, 16.0^\circ$	$\mathbf{0.46m}, 14.9^\circ$	$0.57m, 12.2^\circ$	$0.48m, 10.3^\circ$	$0.50m,\ \mathbf{9.5}^\circ$	$0.51m, 10.4^\circ$	$0.48m, 12.2^\circ$
Complete		$0.40m, 14.8^\circ$	$0.33m, 12.6^\circ$	$0.34m, 11.3^\circ$	$0.28m,\ 8.7^\circ$	$0.26m,\ 7.7^\circ$	$0.28m,\ 7.1^\circ$	$\mathbf{0.20m},\ \mathbf{6.6}^\circ$

Table 1. **Results on 7 scenes dataset:** median position and orientation errors are reported for each scene, best results are in bold. *Scene specific* means that one network was trained for each scene, while *All scenes* is the same method but used with a unique network for all scenes. Method *ICP w/o dc.* relies only on the 3D point position during the point cloud matching. Result in *V + ICP (GT)* have been obtained by using the ground truth depth maps to produce the global point cloud reference (see section 2.4).

Scene	Posenet LSTM [18]	Posenet Geometric [8]	V + ICP (GT) (all scenes)
Chess	$0.24m, 5.77^\circ$	$0.13m, 4.48^\circ$	$0.12m,\ 4.5^\circ$
Fire	$0.34m, 11.9^\circ$	$0.27m, 11.3^\circ$	$0.25m,\ 8.9^\circ$
Heads	$0.21m, 13.7^\circ$	$0.17m, 13.0^\circ$	$0.18m,\ 9.9^\circ$
Office	$0.30m, 8.08^\circ$	$0.19m, 5.55^\circ$	$0.22m,\ 7.3^\circ$
Pumpkin	$0.33m, 7.00^\circ$	$0.26m, 4.75^\circ$	$0.21m,\ 6.2^\circ$
Kitchen	$0.37m, 8.83^\circ$	$0.23m, 5.35^\circ$	$0.15m,\ 4.5^\circ$
Stairs	$0.40m, 13, 7^\circ$	$0.35m, 12.4^\circ$	$0.48m, 12.2^\circ$
Complete	$0.32m,\ 9, 0^\circ$	$0.22m,\ 6.8^\circ$	$0.20m,\ 6.6^\circ$

Table 2. **Comparison with pose regression network (median position and orientation errors):** Posenet requires a dedicated network for each scene, while our method uses the same network.

iterations. We perform standard image augmentation during training (random cropping and colour alteration). Images are rescaled at 224 × 224 pixels during training and testing and our network produces a depth map 4 times smaller than the input image (*i.e.* 56 × 56).

7 scenes dataset. The dataset is composed of various indoor sequences acquired by an RGBD camera with ground truth poses. From the training sequences, we only use RGBD information (without the frame pose) to train our network. For testing, we use training sequences of each scene as reference data (to create the pool of descriptors $\{\mathbf{f}_{I_k}\}_S$ and to generate the point cloud $\mathcal{P}_{ref}$) and only the images of the test sequences as queries.

3.2. Results

Localisation performances of our proposal are presented on table 1. Results show that NetVLAD descriptor produces more accurate result than MAC descriptor and our refinement step significantly increases the final pose precision (M + ICP & V + ICP respectively compared to MAC & NetVLAD). We also show the benefit of using point descriptors computed by our encoder for the point cloud matching described in section 2.4 (M + ICP compared to M + ICP w/o dc.). Finally we compare the performances of a single network trained on all scenes for depth estimation compared to specific network trained on each scene. We observe a slight decrease in pose accuracy for some scenes, but the overall performances remain stable. Best results are obtain by using the ground truth depth maps to build the global reference point cloud (while the depth map related to the query remain created by our encoder-decoder network). These encouraging results confirm our hypothesis that the image-only pose estimation problem can be addressed by one global method instead of multiple systems for each scene where we want to localise a camera.

Comparison with Posenet. We report on table 2 localisation performances of our method trained on all scenes network, compared to Posenet [20, 8]. Posenet learns a mapping from images to 6 DoF camera poses so we need to train a network by scene. We show that our proposal is more accurate than both LSTM Posenet [18] and the last version presented in [8], while using a single network for all scenes, compared to the 7 networks needed for Posenet.

4. CONCLUSION

We have presented a new method for camera relocalisation that benefits from both learned features and geometric information. We show that we are able to refine a pose associated to an image by aligning a point cloud entirely generated from the RGB modality. Our method is generic and can be used on various scenes without specific retraining.

In a future work, we will investigate relocalisation performances of our method on outdoor scenes, as well as a version of our system that can be trained with only RGB data by self-supervised training [16].

Acknowledgments. We would like to acknowledge the French ANR project pLaTINUM (ANR-15-CE23-0010) for its financial support, and the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

5. REFERENCES

[1] Tommaso Cavallari, Stuart Golodetz, Nicholas A. Lord, Julien Valentin, Victor A. Prisacariu, Luigi Di Stefano, and Philip H. S. Torr, "Real-Time RGB-D Camera Pose Estimation in Novel Scenes using a Relocalisation Cascade," *arXiv preprint*, pp. 1–18, 2018.

[2] Torsten Sattler, Bastian Leibe, and Leif Kobbelt, "Efficient & Effective Prioritized Matching for Large-Scale Image-Based Localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. X, no. 1, 2016.

[3] Jamie Shotton, Ben Glocker, Christopher Zach, Shahram Izadi, Antonio Criminisi, and Andrew Fitzgibbon, "Scene coordinate regression forests for camera relocalization in RGB-D images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 2930–2937.

[4] Nathan Piasco, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet, "A survey on Visual-Based Localization: On the benefit of heterogeneous data," *Pattern Recognition*, vol. 74, pp. 90–109, feb 2018.

[5] David Eigen, Christian Puhrsch, and Rob Fergus, "Depth Map Prediction from a Single Image using a Multi-Scale Deep Network," in *Annual Conference on Neural Information Processing Systems (NIPS)*, 2014, pp. 1–9.

[6] Nathan Piasco, Désiré Sidibé, Valérie Gouet-Brunet, and Cédric Demonceaux, "Learning Scene Geometry for Visual Localization in Challenging Conditions," in *IEEE International Conference of Robotics and Automation (ICRA)*, 2019.

[7] Hajime Taira, Masatoshi Okutomi, Torsten Sattler, Mircea Cimpoi, Marc Pollefeys, Josef Sivic, Tomas Pajdla, and Akihiko Torii, "InLoc: Indoor Visual Localization with Dense Matching and View Synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Alex Kendall and Roberto Cipolla, "Geometric loss functions for camera pose regression with deep learning," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] Eric Brachmann and Carsten Rother, "Learning Less is More - 6D Camera Localization via 3D Surface Regression," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[10] Vassileios Balntas, Shuda Li, and Victor Prisacariu, "RelocNet : Continous Metric Learning Relocalisation using Neural Nets," in *IEEE European Conference on Computer Vision (ECCV)*, 2018.

[11] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 5297–5307, 2017.

[12] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus, "End-to-End Learning of Deep Visual Representations for Image Retrieval," *International Journal of Computer Vision (IJCV)*, vol. 124, no. 2, pp. 237–254, 2017.

[13] Filip Radenović, Giorgos Tolias, and Ondej Chum, "Fine-tuning CNN Image Retrieval with No Human Annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2017.

[14] Ali Sharif Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki, "Visual Instance Retrieval with Deep Convolutional Networks," *arXiv preprint*, vol. 4, no. 3, pp. 251–258, 2014.

[15] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow, "Unsupervised Monocular Depth Estimation with Left-Right Consistency," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[16] Reza Mahjourian, Martin Wicke, and Anelia Angelova, "Unsupervised Learning of Depth and Ego-Motion from Monocular Video Using 3D Geometric Constraints," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] François Pomerleau, Francis Colas, and Roland Siegwart, "A Review of Point Cloud Registration Algorithms for Mobile Robotics," *Foundations and Trends in Robotics*, vol. 4, no. 1, pp. 1–104, 2015.

[18] Florian Walch, Caner Hazirbas, Laura Leal-Taixé, Torsten Sattler, Sebastian Hilsenbeck, and Daniel Cremers, "Image-based Localization with Spatial LSTMs," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[19] Phillip Isola, Jun Yan Zhu, Tinghui Zhou, and Alexei A. Efros, "Image-to-image translation with conditional adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 5967–5976.

[20] Alex Kendall, Matthew Grimes, and Roberto Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.