RPNet: an End-to-End Network for Relative Camera Pose Estimation

Sovann En, Alexis Lechervy, and Frédéric Jurie

Normandie Univ, UNICAEN, ENSICAEN, CNRS — UMR GREYC
firstname.lastname@unicaen.fr

Abstract. This paper addresses the task of relative camera pose estimation from raw image pixels, by means of deep neural networks. The proposed RPNet network takes pairs of images as input and directly infers the relative poses, without the need of camera intrinsic/extrinsic. While state-of-the-art systems based on SIFT + RANSAC, are able to recover the translation vector only up to scale, RPNet is trained to produce the full translation vector, in an end-to-end way. Experimental results on the Cambridge Landmark data set show very promising results regarding the recovery of the full translation vector. They also show that RPNet produces more accurate and more stable results than traditional approaches, especially for hard images (repetitive textures, textureless images, *etc.*). To the best of our knowledge, RPNet is the first attempt to recover full translation vectors in relative pose estimation.

Keywords: relative pose estimation · pose estimation · posenet

1 Introduction

In this paper, we are interested in *relative camera pose estimation* — a task consisting in accurately estimating the location and orientation of the camera with respect to another camera's reference system. Relative pose estimation is an essential task for many computer vision problems, such as Structure from Motion (SfM), Simultaneous Localisation And Mapping (SLAM), *etc.* Traditionally, this task can be accomplished by i) extracting sparse keypoints (ex. SIFT, SURF), ii) establishing 2D correspondences between keypoints and iii) estimating the essential matrix using 5-points or 8-point algorithms [13]. RANSAC is very often used to reject outliers in a robust manner.

This technique, although it has been considered as the de facto standard for many years, presents two main drawbacks. First, the quality of the estimation depends heavily on the correspondence assignment. This is to say, too few correspondences (textureless objects) or too many noisy correspondences (repetitive texture or too much viewpoint change) can lead to surprisingly bad results. Second, the traditional method is able to estimate the translation vector only up to scale (directional vector).

In this paper, our objective is three folds: i) we propose a system producing more stable results ii) recovering the full translation vector iii) and we provide insights regarding relative pose inference (*i.e.* from absolute pose, regressor *etc.*).

As pointed out in [20], CNN based methods are able to produce pretty good results in some cases where SIFT-based methods fail (*i.e.* texture less images). This is the reason why we opted for a global method based on CNN. Inspired by the success of PoseNet [9], we propose a modified Siamese PoseNet for relative camera pose estimation, dubbed as RPNet, with different ways to infer the relative pose. To the best of our knowledge, [12] is the only end-to-end system aiming at solving relative camera pose using deep learning approach. However, their system estimate the translation vector up to scale, while ours produces full translation vectors.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 introduces the network architecture and the training methodology. Section 4 discusses the datasets and presents the experimental validation of the approach. Finally, Section 5 concludes the paper.

2 State of the Art

Local keypoint-based approaches. They address relative camera pose estimation using the epipolar geometry between 2D-2D correspondences of keypoints. Early attempts aimed at better engineering interest point detectors to focus on interesting image properties such as corners [6], blobs in scale-space [10], regions [11], or speed [2, 18, 16] *etc.* More recently, there is a growing interest to train interest point detectors together with the matching function [5, 23, 17, 4, 19]. LIFT [21] adopted the traditional pipeline combining a detector, an orientation estimator, and a descriptor, tied together with differentiable operations and learned end-to-end. [1] proposed a multitask network with different sub-branches to operate on varying input sizes. [4] proposed a bootstrapping strategy by first learning on simple synthetic data and increasing the training set with real images in a second time.

End-to-End pose estimation. The first end-to-end neural network for camera pose estimation from single RGB images is PoseNet [9]. It is based on GoogLeNet with two output branches to regress translations and rotations. PoseNet follow-up includes: Baysian PoseNet [7], Posenet-LSTM [20] where LSTM is used to model the context of the images, Geometric-PoseNet where the loss is calculated using the re-projection error of the coordinates using the predicted pose and the ground truth [8]. Since all the 3D models used for comparisons are created using SIFT-based techniques, traditional approach seems more accurate. [20] showed that the classical approaches completely fail with less textured datasets such as the proposed TMU-LSI dataset. [14] is an end-to-end system for pose regression taking sparse keypoint as inputs. Regarding relative pose estimation, [12] is the only system we are aware of. Their network is based on ResNet35 with FCs layers acting as pose regressor. Similar to the previous networks, the authors formulate the loss function as minimising the L2-distances between the ground truth and the estimated pose. Unfortunately, several aspects of their results (including their label generation, experimental methodology and the baseline system) make comparisons difficult. Along side with pose regression problems, another promis-

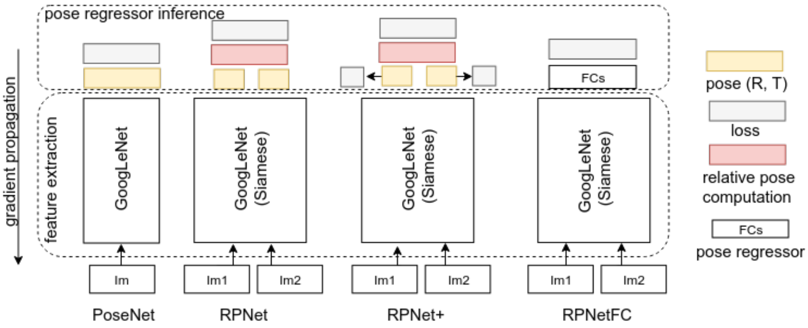

Fig. 1. Illustration of the proposed system

ing works from [15] showed that an end-to-end neural network can effectively be trained to regress to infer the homography between two images. Finally, two recent papers [22, 3] made useful contributions to the training of end-to-end systems for pose estimation. [22] proposed a regressor network to produce essential matrix which can be then used to find the relative pose. However, their system is able to find the translation up to scale which is completely different from our objective. In [3], a differentiable RANSAC is proposed for outlier rejection and can be a plug-and-play component into an end-to-end system.

3 Relative pose inference with RPNet

Architecture. The architecture of the proposed RPNet, illustrated Fig. 1, is made of two building blocks: i) a Siamese Network with two branches regressing one pose per image, ii) a pose inference module for computing the relative pose between the cameras. We provide three variants of the pose inference module: (1) a parameter-free module, (2) a parameter-free module with additional losses (same as PoseNet loss [9]) aiming at regressing the two camera poses as well as the relative pose, and (3) a relative pose regressor based on FC layers. The whole network is trained end-to-end for relative pose estimation. Inspired by PoseNet [9], the feature extraction network is based on the GoogLeNet architecture with 22 CNN layers and 6 inception modules. We only normalize the quaternion during test time. It outputs one pose per image.

For RPNet and RPNet$^+$, the module for computing the relative pose between the cameras is straightforward and relies on simple geometry. Following the convention of OpenCV, the relative pose is calculated in the reference system of the 2nd camera. Let (R_1, t_1, R_2, t_2) be the rotation matrices and translation vectors used to project a point X from world coordinate system to a fixed camera system (camera 1 & 2). Let (q_1, q_2) be the corresponding quaternions of (R_1, R_2). The relative pose is calculated as followed:

$$R_{1,2} = q_2 \times q_1^* \quad \text{and} \quad T_{1,2} = R_2(-R_1^T t_1) + t_2 \tag{1}$$

Table 1. Number of training and testing pairs for Cambridge Landmark dataset. SE stands for spatial extent, measured in meter.

Scene	Train	Test	SE	Scence	Train	Test	SE
Kings College	9.1k	2.4k	140x40	Shop Facade	1.6k	0.6k	35x25
Old Hospital	6.5k	1.2k	50x40	St Marys Church	11k	4.1k	80x60

where q_1^* is the conjugate of q_1, and × denotes the multiplication in the quaternion domain. Both equations are differentiable. For RPNetFC, the pose inference module is a simple stacked fully connected layers with *relu* activation. To limit over-fitting, we modified the output of the Siamese network by reducing its output dimension from 2048 to 256. This results in almost 50% reduction of the number of parameters compared to PoseNet, RPNet and RPNet$^+$ network. The pose regressor network contains two FC layers (both with 128 dimensions).

Losses. The loss function uses the Euclidean distance to compare predicted relative rotation $q\hat{_{1,2}}$ and translation $\hat{T}_{1,2}$ with ground truth $\hat{q}_{1,2}$ and $q_{1,2}$: $loss = \sum_i(||\hat{T}^i_{1,2} - T^i_{1,2}||_2 + \beta * ||\hat{q}^i_{1,2} - q^i_{1,2}||_2)$. Quaternions are unit quaternions. The original PoseNet has a β term in front of quaternions to balance the loss values between the translation and rotation. To find the most suitable value of β, we cross-validated on our validation set. Please refer to our codes for different hyper-parameter values on different subsets.

4 Experimentations

4.1 Experimental Setup

Dataset. Experimental validation is done on the Cambridge Landmark dataset[1]. Each image is associated with a ground-truth pose. We provide results on 4 of the 5 subsets (scenes). As discussed by several people, the 'street' scene raises several issues[2].

Pair generation. For each sequence of each scene, we randomly pair each image with eight different images of the same sequence. For a fair comparison with SURF, the pair generation is done by making sure that they overlap enough. We followed the train-test splits defined with the data set. Images are scaled so that the smallest dimension is 256 pixels, keeping its original aspect ratio. During training, we use 224*224 random crops and feed them into the network. During test time, we center crop the image.

Baseline. The baseline is a traditional keypoint-based method (SURF). The focal length and the principle point are provided by the dataset. Other parameters are cross-validated on the validation set. For a fair comparison, we provide two scenarios for baselines: (1) the image are scaled to be 256*455 pixels, followed by a center-crop (224*224 pixels) to produce the same image pairs as tested with our networks and (2) the original images without down-sampling. We named

[1] `http://mi.eng.cam.ac.uk/projects/relocalisation`

[2] `https://github.com/alexgkendall/caffe-posenet/issues/2`

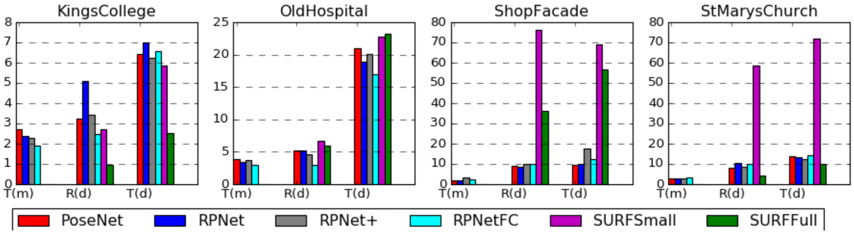

Fig. 2. Translation and Rotation errors (median) of the different approaches

these two scenarios as 'SURFSmall' and 'SURFFull'. All the camera parameters are adapted to the scaling and cropping we applied.
Evaluation metric. We measured 3 different errors: i) translation errors, in meters ii) rotation errors, in degrees and iii) translation errors in degrees. We report the median for all the measurements.

4.2 Experimental results

Relative pose inference module. Fig. 2 compares the performance of the different systems and test scenarios. Based on these experimental results, RPNetFC and RPNet$^+$ are the most efficient ways to recover the relative pose. On easy dataset (*i.e.*. KingsCollege and OldHospital), where there is no ambiguity textures, using pose regressor (RPNetFC) produces slightly better results than inferring the relative pose from the two images (PoseNet/RPNet/RPNet$^+$). On the contrary, on hard datasets (*i.e.*. ShopFacade and StMarysChurch), RPNet-family outperforms RPNetFC. This behavior is also true for relative rotation and relative translation measured in degree. Globally, RPNetFC produces the best results followed by RPNet$^+$, PoseNet and finally, RPNet. The differences of their results are between 0 and 8 degrees. Regarding technical aspect, RPNetFC is a lot easier to train than RPNet$^+$/RPNet since it does not involve multiple hyper-parameters to balance the different losses. It also converges faster.
Comparison with traditional approaches. We will start by discussing the SURFSmall scenario first. In general, the error on both translation and rotation can be reduced between 5 to 70% using RPNet family, except on KingsCollege where the traditional approach slightly outperforms RPNet-based methods. We observed that the performance of the traditional approaches varies largely from one subset to another, while RPNet$^+$/RPNetFC are more stable. In addition, the traditional approach requires camera information for each image in order to correctly estimate the pose. In contrast, RPNet-based does not require any specific information at all. Using the original image size (SURFFull) significantly boost the performance of the traditional approach. However, RPNetFC still enjoy a significant gain in performance on OldHospital and ShopFacade, while performing slightly worse than SURFFull on KingsCollege and StMarysChurch. The difference in performance between SURFFull and RPNetFC is even more significant when the images contain large view point changes (see Fig. 3).

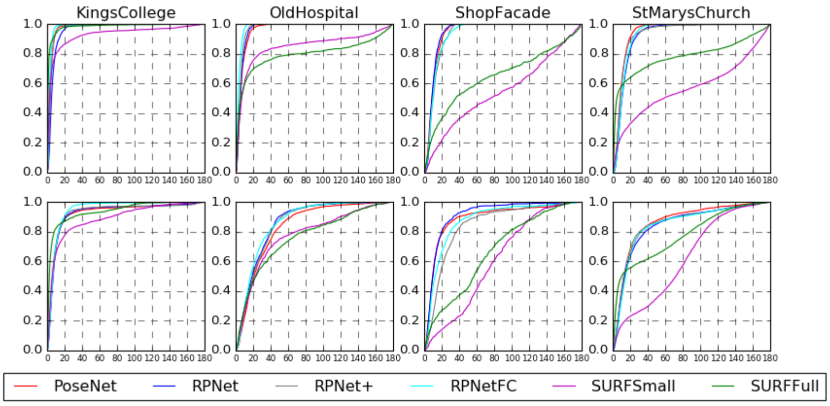

Fig. 3. Accumulative hist. of errors in rotation (1st row, d), translation (2nd row, m).

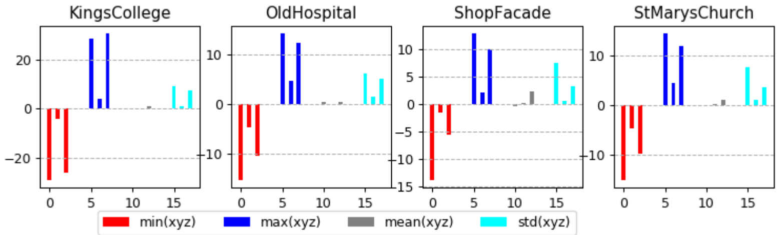

Fig. 4. Min/Max/Mean/STD relative translations (ground truth), w.r.t. XYZ axis (m).

Full translation vector. One of our objectives is to provide a system able to estimate the full translation vector. On average, we observed that the median error ranges between 2 to 4 meters, using RPNetFC. Fig. 4 gives an idea of ground truth translations w.r.t. reference axes (xyz). For instance, on KingsCollege, the values of X-axis can range from -29m to 30m with an STD of 7 meters. Interestingly, our network has a translation error of only 2.88 meters.

5 Conclusions

This paper proposed a novel architecture for estimating full relative poses using an end-to-end neural network. The network is based on a Siamese architecture, which was experimented with different ways to infer the relative poses. In addition, to produce competitive or better results over the traditional SURF-based approaches, our system is able to produce an accurate full translation vector. We hope this paper will provide more insight and motivate other researchers to focus on global end-to-end system for relative pose regression problems.

Acknowledgements. This work was partly funded by the FrenchUK MCM ITP program and by the ANR-16-CE23-0006 program.

References

1. Altwaijry, H., Veit, A., Belongie, S.J.: Learning to detect and match keypoints with deep architectures. In: Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016 (2016)
2. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: European conference on computer vision. pp. 404–417 (2006)
3. Brachmann, E., Krull, A., Nowozin, S., Shotton, J., Michel, F., Gumhold, S., Rother, C.: Dsac-differentiable ransac for camera localization. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). vol. 3 (2017)
4. DeTone, D., Malisiewicz, T., Rabinovich, A.: Superpoint: Self-supervised interest point detection and description. arXiv preprint arXiv:1712.07629 (2017)
5. Han, X., Leung, T., Jia, Y., Sukthankar, R., Berg, A.C.: Matchnet: Unifying feature and metric learning for patch-based matching. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. pp. 3279–3286 (2015)
6. Harris, C.G., Stephens, M.: A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference, AVC 1988, Manchester, UK, September, 1988. pp. 1–6 (1988)
7. Kendall, A., Cipolla, R.: Modelling uncertainty in deep learning for camera relocalization. In: Robotics and Automation (ICRA), 2016 IEEE International Conference on. pp. 4762–4769 (2016)
8. Kendall, A., Cipolla, R.: Geometric loss functions for camera pose regression with deep learning. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 6555–6564. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.694
9. Kendall, A., Grimes, M., Cipolla, R.: Posenet: A convolutional network for real-time 6-dof camera relocalization. In: Computer Vision (ICCV), 2015 IEEE International Conference on. pp. 2938–2946 (2015)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
11. Matas, J., Chum, O., Urban, M., Pajdla, T.: Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing **22**(10), 761–767 (2004)
12. Melekhov, I., Ylioinas, J., Kannala, J., Rahtu, E.: Relative camera pose estimation using convolutional neural networks. In: International Conference on Advanced Concepts for Intelligent Vision Systems. pp. 675–687 (2017)
13. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE transactions on pattern analysis and machine intelligence **26**(6), 756–770 (2004)
14. Purkait, P., Zhao, C., Zach, C.: Spp-net: Deep absolute pose regression with synthetic views. arXiv preprint arXiv:1712.03452 (2017)
15. Rocco, I., Arandjelovic, R., Sivic, J.: Convolutional neural network architecture for geometric matching. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 39–48. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.12
16. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: Computer Vision (ICCV), 2011 IEEE international conference on. pp. 2564–2571 (2011)
17. Tian, Y., Fan, B., Wu, F., et al.: L2-net: Deep learning of discriminative patch descriptor in euclidean space. In: Conference on Computer Vision and Pattern Recognition (CVPR). vol. 2 (2017)

18. Tola, E., Lepetit, V., Fua, P.: Daisy: An efficient dense descriptor applied to wide-baseline stereo. IEEE transactions on pattern analysis and machine intelligence **32**(5), 815–830 (2010)
19. Trujillo, L., Olague, G.: Using evolution to learn how to perform interest point detection. In: Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. vol. 1, pp. 211–214 (2006)
20. Walch, F., Hazirbas, C., Leal-Taixé, L., Sattler, T., Hilsenbeck, S., Cremers, D.: Image-based localization using lstms for structured feature correlation. In: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. pp. 627–637. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.75
21. Yi, K.M., Trulls, E., Lepetit, V., Fua, P.: Lift: Learned invariant feature transform. In: European Conference on Computer Vision. pp. 467–483 (2016)
22. Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). vol. 3 (2018)
23. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. pp. 4353–4361 (2015)