HAL
open science

Vehicle absolute ego-localization from vision, using only pre-existing geo-referenced panoramas

Fabien Moutarde, Guillaume Bresson, Yu Li, Cyril Joly

To cite this version:

Fabien Moutarde, Guillaume Bresson, Yu Li, Cyril Joly. Vehicle absolute ego-localization from vision, using only pre-existing geo-referenced panoramas. Reliability and Statistics in Transportation and Communications, Oct 2019, Riga, Latvia. hal-02342259

HAL Id: hal-02342259

https://minesparis-psl.hal.science/hal-02342259v1

Submitted on 31 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Vehicle absolute ego-localization from vision, using only pre-existing geo-referenced panoramas

Fabien Moutarde[1], Guillaume Bresson[2], Li Yu[1,2] and Cyril Joly[1]

[1]Center for Robotics, MINES ParisTech, PSL Université, 60 bd St Michel, 75006 Paris, France
Firstname.Name@mines-paristech.fr

[2] Institut VEDECOM, 23 bis Allée des Marronniers, 78000 Versailles, France
firstname.name@vedecom.fr

Abstract. Precise ego-localization is an important issue for Intelligent Vehicles. Geo-positioning with standard GPS often has localization error up to 10 meters, and is even sometimes unavailable due to "urban canyon" effect. *It is therefore an interesting goal to design an affordable and robust alternative to GPS ego-localization*. In this paper, we present 2 approaches for absolute ego-localization based *on vision only*, and not requiring previous driving on same street, by leveraging only pre-existing geo-referenced panoramas such as those from Google StreetView. Our first variant is based on Bag of visual Words + visual keypoints matching + bundle adjustment, and the other one uses direct pose regression computed by a deep Convolutional Neural Network (CNN) taking the query image as input. We have evaluated our 2 proposed variants using a real car. On around 1 km in a dense urban area, we obtained average localization errors of 2.8m with visual keypoints-matching + geometric computations, and of 7.7m with pose regression using pre-trained deep CNN. This shows that our proposed approaches are therefore potentially interesting complements or even alternatives to GPS localization.

Keywords: Intelligent Vehicle, ego-localization, visual localization, place visual recognition

1 Introduction

Self-driving cars, and Intelligent Vehicles in general, have made tremendous progresses in the last decade. By combining many sensor types (cameras, radars, lidars, GPS, etc…) to perceive the surroundings, and using several smart algorithms (such as Deep-Learning) to perform semantic interpretation of the raw data of sensors, an on-board program can localize road, lanes and obstacles in order to drive autonomously, or at least provide valuable Advanced Driving Assistance Systems (ADAS) such as Lane Keeping and Adaptive Cruise Control.

However, due to their complexity, urban environments remain challenging for those systems. Moreover, because of frequent intersections, it is particularly important in those contexts to estimate precisely (i.e. ideally with precision < 1 meter) the absolute ego-localization of the vehicle. GPS does usually provide absolute ego-localization. But geo-positioning with standard GPS often has localization error up to 10 meters, and unfortunately, GPS localization precision is particularly bad in urban areas, and is even sometimes totally unavailable due to the "urban canyon" effect (Drawil et al. 2012).

An alternative geo-localization approach consists in using integration of some odometry based either on wheels, or on vision or lidar. However, this is unreliable on long

distance, as odometry-based localization is bound to drift due to error accumulation along trip (Zhang & Singh S., 2015). As for precise-enough inertial reference system, they are generally too expansive for vehicles. *It is therefore an interesting goal to design an affordable and robust alternative to GPS ego-localization*.

In this paper, we propose an approach for absolute ego-localization based *on vision only*, and *not requiring previous driving on same street*: we show that it is possible to obtain GPS-level precision (few meters) of localization by *leveraging only pre-existing geo-referenced panoramas* such as those from Google StreetView (Anguelov 2010).

2 Related work

Ego-localization has been the subject of many scientific researches. Two recent surveys of this topic (Bresson et al. 2017)(Cadena et al. 2016) can be referred to for a global view of these researches. We focus here only on *vision-based* ego-localization systems using an existing source of information.

Such visual-map aided ego-localization systems are very few in the literature and mainly use Google Street View panoramas, due to the presence of their accurate positioning and of a coarse depth map (see Fig. 2 and Fig.3 for an example). The problem is actually mostly addressed with a *place recognition* objective. In (Madjik et al. 2013) a ground-air place recognition system is proposed, that matches aerial images with StreetViews and 3D cadastral building models. Street Views are converted into a feature-based representation using Affine Scale-Invariant Feature Transform (ASIFT). (Zamir and Shah 2010) use another type of features: they build an indexed tree based on SIFT descriptors extracted from 100,000 Street Views, and then choose the panorama closest to the query image by applying a voting scheme. From a place recognition point of view, the main challenge remains to find descriptors that are informative enough at a whole city scale (Schindler et al. 2007) (Torii et al. 2011) (Baatz et al. 2012).

As for visual *metric* localization, one of oldest works is that of (Zhang and Kosecka 2006) in which the position of the camera is estimated by triangulation between several geo-referenced Street Views. More recently, (Agarwal et al. 2015) have proposed a two-stage approach for ego-localization based on StreetView: in the first phase, the 3D positions of tracked features in monocular sequences are estimated; then, an association of these points with StreetViews is used to compute a relative transformation. Other approaches involving pre-existing data other than StreetView can also be found such as the use of aerial images (Kummerlemmerle et al. 2011), or the integration of geo-referenced objects (traffic lights and signs, for instance) to constrain the localization (Qu et al. 2015).

The rise of deep learning in perception has led to totally new approaches regarding localization. The seminal PoseNet work (Kendal et al. 2015) is an approach that uses a Convolutional Neural Network (CNN) to directly regress from a query image the corresponding 6 DoF pose. The CNN is trained on custom datasets of images with associated poses generated using Structure from Motion (SfM). The obtained ego-localization accuracy is only of a few meters (between 1.5 to 3.7 m depending on the validation test) but the pose regression CNN exhibits good robustness to common image appearance

changes (illumination, weather, and presence/absence of non-static objects), and requires less computational time that a standard SfM approach. (Clark et al. 2017) have later proposed VidLoc, an approach using the same principle as PoseNet, but taking into account the temporal link between images using an LSTM (Long Short-Term Memory) Recurrent Neural Network, which significantly improves the results of PoseNet. However, the two above works required extensive prior image recordings close to the path of future online ego-localization. Very recently (Mirowski et al. 2019) showed that it is possible to learn how to navigate in multiple cities, after prior training with Deep Reinforcement Learning using StreetView information. Their work, however, does not include explicit metric localization.

3 Proposed methods

Our 2 proposed methods work in two phases. The first phase is an offline step in which Street View panoramas, along with their depth maps and absolute positions, located in the test area are extracted. Panoramas are transformed into a set of rectilinear images similar in constitution to the images that will be processed on-board during online localization. This part is common to our 2 variants, and detailed below.

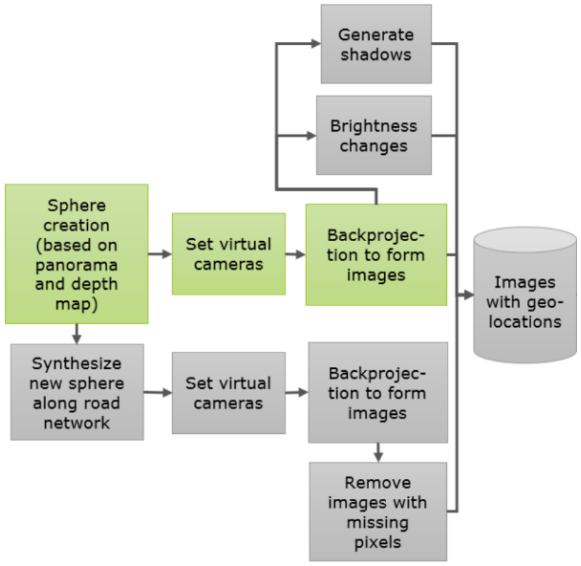

Fig. 1. Pipeline for offline generation of geo-referenced rectilinear images from each StreetView panorama.

3.1 Offline pre-processing and augmentation of StreetView panoramas

The whole pipeline is illustrated in Fig. 1. From each StreetView panorama, we first create a set of rectilinear images similar to the images that shall be used on-board the car for online localization (green boxes). These rectilinear views are obtained from the 360° panorama by creating several virtual pinhole cameras located at the centre O of

the panorama, and using the intrinsic calibration matrix K of the target camera (the one that will be on-board the car during online ego-localization). A typical example of such set of rectilinear images is shown on Fig. 2.

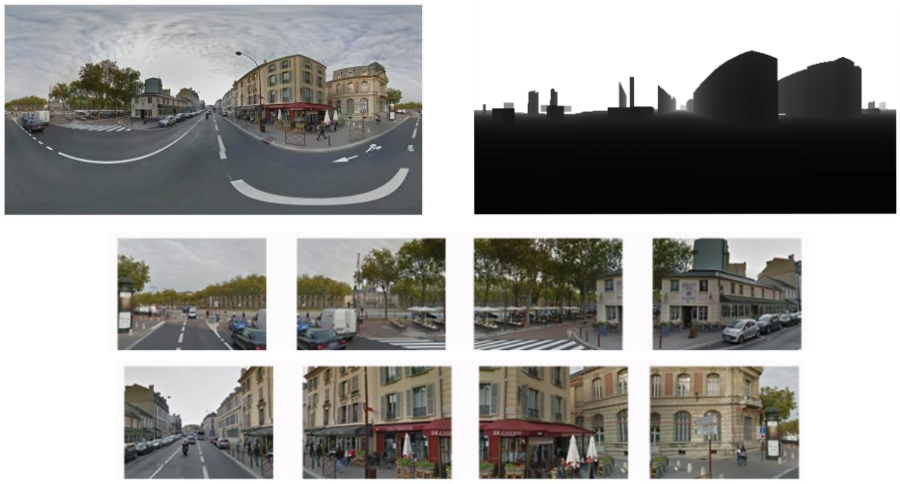

Fig. 2. On top, example of StreetView 360° panorama (upper-left) and its associated depth map (upper-right). On bottom, examples of generated rectilinear images with various orientations of viewpoint from the same point O (center of the source panorama).

In the area used in the experiments, StreetView panoramas are distributed along the road network with an average distance of 6 to 16 meters. This is clearly not dense enough to hope obtaining a longitudinal precision of a few meters in final localization. We therefore augment the initial database by generating synthetic rectilinear images from several virtual points O' located at various intermediate position between existing panoramas (grey boxes). To do so, we use the depth map associated to each panorama, to build a back-projection model using ray tracing and bilinear interpolation, as proposed in (Meilland et al. 2010), and illustrated on Fig. 3. We generate images following the direction of the road within a 4-meter range and with a 0.2-meter step, resulting in 40 new locations from which to synthesize images for each panorama. More details on the mathematical formulas can be found in (Yu et al. 2016b).

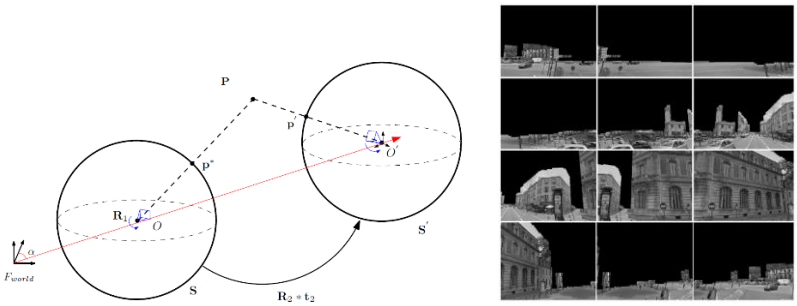

Fig. 3. Generation of rectilinear images from virtual viewpoints located in point O' translated from the centre O of the initial panorama. Black regions correspond to pixels for which the back-projection lies outside the original panorama, and is therefore unknown.

3.2 Method 1: Image keypoints, BoW, and geometry

Our first variant of visual ego-localization, illustrated by the pipeline of Fig. 4, is purely based on image analysis and geometry. From each query image captured by the on-board camera, we first determine an approximate location by a "bag of visual words" (BoW) method returning best match from our set of pre-generated rectilinear images. We then further refine the localization estimate by applying a Local Bundle Adjustment. More details on the algorithm can be found in (Yu et al. 2016a). As can be seen on Fig. 4, in order to reduce online computations, the keypoints and BoW are precomputed offline on all rectilinear images of our dataset.

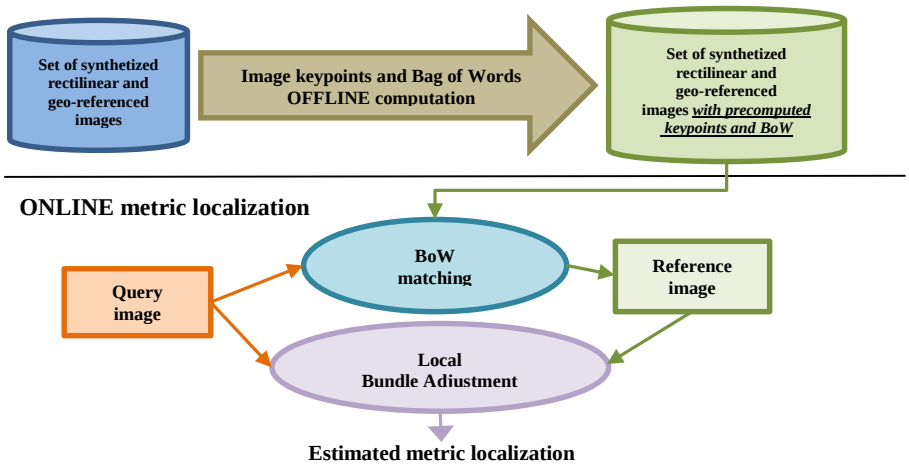

Fig. 4. Pipelines for our method 1: on top, keypoints and BoW are pre-computed *offline* on all synthetized geo-referenced images; on bottom, the 2 steps for online metric localization.

3.3 Method 2: Pose regression by Convolutional Neural Network (CNN)

Our second variant of visual ego-localization uses the principle of direct pose regression from query image by a Convolutional Neural Network, following the idea of PoseNet (Kendall et al. 2015). The original PoseNet is based on a slightly modified version of GoogleNet where the final softmax classifier layer is replaced by affine regressors. A Fully Connected layer is added before the 2 regressors (one for estimation of the 3D position of the camera x, and the other one for its orientation under the form of a quaternion q). Stochastic Gradient Descent is used to train the CNN with the following loss function:

$$L = \|\hat{x} - x\|^2 + \beta\|\hat{q}_q\|^2 \qquad (1)$$

where $\hat{x}$ and $\hat{q}$ are the regressed estimations, x and q are the ground truth, and the parameter β is used to adjust the relative weight between position and orientation errors and can be fine-tuned with grid search.

While in PoseNet, the training was performed on very dense prior image recordings conducted close to the path of future online ego-localization, in our case the training images are only the rectilinear images generated from pre-existing StreetView panoramas, as described in §3.1. Furthermore, we modified the PoseNet approach in 2 ways:

1/ we simplified the regressor outputs in order to provide only a 2D position x_{2D} and a global orientation θ; 2/ we changed the CNN architecture from GoogleNet to ResNet50. Note that our regressed 2D positions are projected from latitude and longitude to Universal Transverse Mercator (UTM), and that we center them on the mean position of the Street View panoramas of the test area, in order to reduce the magnitude of the values that are regressed by the CNN. Due to our first modification, the loss function minimized during training is changed to:

$$L = \|\widehat{x_{2D}} - x_{2D}\|^2 + \beta\left\|\hat{\theta} - \theta\right\|^2 \qquad (2)$$

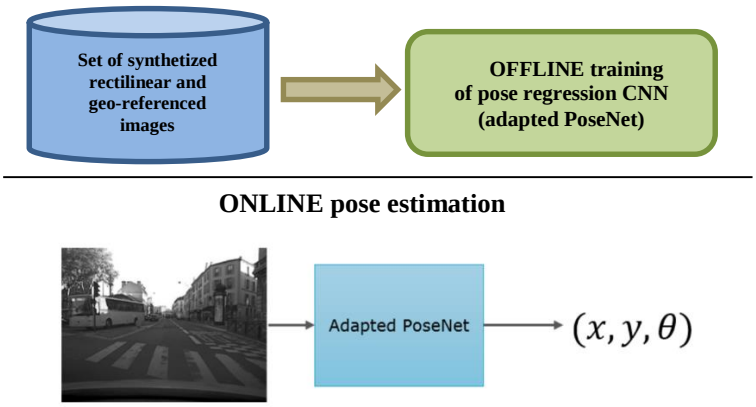

Fig. 5. Pipelines for our method 2: on top, a pose regression CNN is pre-trained *offline* on the dataset of all synthetized geo-referenced images; on bottom, the straightforward pose estimation computed directly from the query image.

As for training, similarly to PoseNet approach, we use transfer learning to initialize the weights of convolutional layers with values from the original ResNet50 trained for classification on ImageNet. Note that every image from our set of rectilinear images was resized to a 224×224 resolution, in order to fit the CNN input. Training was done by Stochastic Gradient Descent using the Adam optimizer with a learning rate of 10^{-5} and a batch size of 80 samples during 500 epochs.

4 Experiments and results

The 2 presented methods were evaluated using several acquisitions made in the city of Versailles, France. The vehicle was equipped with a camera, and a Real Time Kinematic GPS fused with a high-end Inertial Measurement Unit used only for ground truth purposes (accuracy of a few centimeters). Two different camera settings were tested: a camera facing forward, located inside the vehicle behind the windshield and a camera facing sideway towards building façades (see on top of Fig. 6). The camera provides grayscale images (resolution of 640×480) at 20 Hz. Two examples of images taken from the acquisitions are visible at the bottom of Fig. 6. Note that only images were used to perform the ego-localization estimates by the two methods, and without any

position tracking between two consecutive images (each frame is treated independently of the previous one).

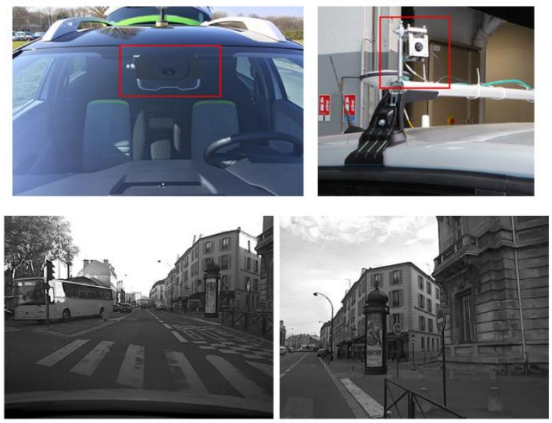

Fig. 6. On top, positions of the cameras in the vehicle used in the experiments. On bottom, examples of acquired images with the corresponding cameras.

We evaluated our 2 methods over 5 trajectories, including 1 with the camera facing forward (denoted as sequence F). The results are reported in Table 1, where 'Fail' indicates, for the features+geometry variant that it could not recognize the place due to many potential candidates (environment not distinctive enough), and for the CNN pose regression variant that the training of the CNN was unable to properly converge (over-fitting or under-fitting observed by a validation dataset excluded from the training).

Table 1. Ego-localization results obtained with our 2 approaches

SeqID (length)	Nb of images	Nb of StView panoramas (nb of *virtual* ones)	Average localization errors	
			image features + geometry	pose regression CNN
1 (234 m)	897	29 (1160)	2.85 m	7.62 m
2 (271 m)	898	29 (1160)	2.63 m	7.93 m
3 (222 m)	895	29 (1160)	Fail	Fail
4 (216 m)	901	34 (1360)	2.82 m	7.55 m
F (265 m)	554	29 (1160)	Fail	7.87 m

As can be seen in Table 1, the localization average error by CNN direct pose regression is around 7.5 to 8 meters. This is larger that our ideal goal (< 1m), but is of the same order of precision as a standard GPS. Our image-features+geometry variant obtains a significantly better accuracy, with average errors ranging from 2.5 to 3 meters. The lower accuracy of CNN regression is probably caused by an insufficient amount of information in the generated images due to missing pixels.

It can be noted that both approaches fail to provide a proper localization in sequence #3. In both cases, we suspect it to be caused by dense vegetation (trees, bushes, etc.) which covers up most of the building façades where distinctive information is usually

found. As for sequence F, with the camera facing forward, only the CNN pose regression variant worked properly, and reached an accuracy similar to that on other sequences. Features were not distinctive enough to obtain a localization with the image-features+geometry variant. Sequences 1 and F are taken in the same area, thus illustrating that CNNs might offer better robustness to the position and orientation of the camera in the vehicle. More detailed evaluations, in particular regarding the respective impacts of our data augmentation process and of our modification from PoseNet can be found in (Bresson et al. 2019).

Finally, regarding computational time, with the appropriate hardware (i.e. a GPU-equipped computer), our CNN approach takes approximately 75 ms per image (~13 frames per seconds) while the features+geometry approach takes 3 seconds on average to compute a position. This makes the CNN regressor approach 40 times faster, and more clearly suitable in practice for real-time on-board use. However, it is worth noting that some parts of the processing in method 1 could be parallelized to improve the overall computational time of this approach.

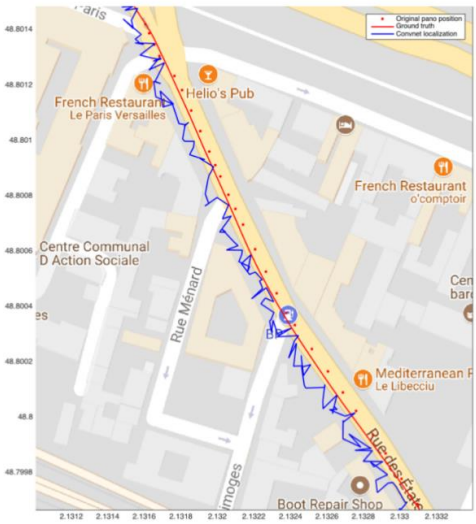

Fig. 7. Example of estimated positions with our pose regression CNN approach (blue line), to be compared with the ground truth trajectory (red line). The red dots are the positions of pre-existing StreetView panoramas.

An example of positions estimated by our CNN pose regressor is plotted in Fig. 7 (blue line), together with the ground truth trajectory (red line) measured by RTK GPS + inertial unit. It can be seen that the trajectory jumps a lot between successive frames, and seems affected by a lateral offset. Jumps were expected as ego-localization is performed independently for each query image, with nothing enforcing the temporal continuity of the localization, and no smoothing post-processing. The lateral offset might be caused by an imbalanced training set as parts of the streets are more represented due the presence of depth information. Even if synthesized images with a majority of zero pixels are discarded from the training set, missing pixels could still have an impact on

how the CNN interprets query images at positions where the training set mainly contained images with many “missing” pixels.

5 Conclusions and perspectives

We have presented approaches for absolute metric ego-localization based *on vision only*, and *not requiring previous driving on same street, by leveraging only pre-existing geo-referenced panoramas from Google StreetView* as pre-requisite input. Our method firstly generates, from each geo-referenced panorama, several synthetic rectilinear images with the same characteristics as the target on-board camera. From these, we tried two very different approaches for estimating localization from on-board query image: 1/ a first variant uses visual keypoints for Bag of Word matching, followed by relative pose estimation performed with Local Bundle Adjustment; 2/ a second variant relies on direct pose regression computed by a deep Convolutional Neural Network (CNN) pre-trained offline on the whole dataset of geo-referenced images.

We have evaluated our 2 methods using a real car, equipped with a monocular camera and a differential RTK GPS providing centimetric precision for position ground truth, and driving around 1 km in a dense urban area. The obtained average localization errors were respectively 2.8m with our image-feature+geometry variant and 7.7m with pose regression using pre-trained deep CNN. These accuracies are both comparable to the precision of a standard GPS, and could therefore potentially be interesting complements. or even alternatives to GPS localization, in order to mitigate its well-known low accuracy or even unavailabity in dense urban areas due to "urban canyon" effect. Furthermore, even if the accuracy of CNN pose regression seems significantly lower, it is on the other hand 40 times faster during online localization, reaching real-time capability (13 frames per seconds).

As future work, our methods could be further improved, firstly by enforcing more continuity between successive ego-localization, either by simple smoothing, or by adding a Recurrent Neural Network. The final outcome could also be improved by better synthesis of images for viewpoints between pre-existing panoramas, either by using some kind of interpolation of successive panoramas, and/or by filling unknown parts with a Generative Adversarial Network (GAN).

Acknowledgements

This work was jointly supported by the Institut VEDECOM of France under the autonomous vehicle project, and the China Scholarship Council (CSC).

References

1. Agarwal P., Burgard W., and Spinello L.: *Metric Localization using Google Street View*. Computing Research Repository (2015).
2. Anguelov D., Dulong C., Filip D., et al.: *Google street view: Capturing the world at street level*. Computer, *43*(6), 32-38 (2010).
3. Baatz G., Köser K., Chen D., Grzeszczuk R., and Pollefeys M.: *Leveraging 3D city models for rotation invariant place-of-interest recognition*. International Journal of Computer Vision, 96(3):315–334 (2012).

4. Bresson G., Alsayed Z., Yu L., and Glaser S.: *Simultaneous Localization And Mapping: A Survey of Current Trends in Autonomous Driving*. IEEE Transactions on Intelligent Vehicles, 2(3) (2017).
5. Bresson G., Li Y., Joly C. and Moutarde F.: *Urban Localization with Street Views using a Convolutional Neural Network for End-to-End Camera Pose Regression*. In: *2019 IEEE Intelligent Vehicles Symposium (IV 19)*, Paris (2019).
6. Cadena C., Carlone L., Carrillo H., Latif Y., Scaramuzza D., Neira J., Reid I., and Leonard J.J.: *Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age*. IEEE Transactions on Robotics, 32(6):1309–1332 (2016).
7. Clark R., Wang S., Markham A., Trigoni N., and Wen H.: *VidLoc: A Deep Spatio-Temporal Model for 6-DoF Video-Clip Relocalization*. In: IEEE Conference on Computer Vision and Pattern Recognition (2017).
8. Drawil N. M., Amar H. M., & Basir O. A.: *GPS localization accuracy classification: A context-based approach*. In proc. of *IEEE* Transactions on Intelligent Transportation Systems, 14(1), 262-273 (2012).
9. Kendall A., Grimes M., and Cipolla R.: *PoseNet: A convolutional network for real-time 6-DOF camera relocalization*. In proc. of IEEE Int. Conf. on Computer Vision (ICCV'2015), pages 2938–2946 (2015).
10. Kummerlemmerle R., Steder B., Dornhege C., Kleiner A., Grisetti G. and Burgard W.:. *Large scale graph-based SLAM using aerial images as prior information*. Autonomous Robots, 30(1):25–39 (2011).
11. Majdik A., Albers-Schoenberg Y., and D. Scaramuzza D.: *MAV Urban Localization from Google Street View Data*. In IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3979–3986 (2013).
12. Meilland M., Comport A.I., and Rives P.: *A Spherical Robot-Centered Representation for Urban Navigation*. In proc. Of IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 5196–5201 (2010).
13. Mirowski P., Grimes M. K., Malinowski M., Hermann K. M., Anderson K., Teplyashin D., Simonyan K., Kavukcuoglu K., Zisserman A., and R. Hadsell R.: *Learning to Navigate in Cities Without a Map*. CoRR,abs/1804.00168 (2018).
14. Qu X., Soheilian B., and Paparoditis N.: *Vehicle localization using mono-camera and geo-referenced traffic signs*. In: IEEE Intelligent Vehicles Symposium, pages 605–610, 2015.
15. Schindler G., Brown M., and R. Szeliski R.: *City-scale location recognition*. In: IEEE Conference on Computer Vision and Pattern Recognition, pages 1–7. (2007).
16. Torii A., Sivic J., and Pajdla T.: *Visual localization by linear combination of image descriptors*. In: IEEE International Conference on Computer Vision Workshops, pages 102–109 (2011).
17. Yu L., Joly C., Bresson G., and Moutarde F.: *Monocular Urban Localization using Street View*. In proc. of 14th Int. Conf. on Control, Automation, Robotics and Vision (ICARCV'2016), pages 1–6, (2016)
18. Yu L., Joly C., Bresson G., and Moutarde F.: *Improving Robustness of Monocular Urban Localization using Augmented Street View".* In proc. of 19th IEEE Int. Conf. on Intelligent Transportation Systems (ITSC'2016), Rio de Janeiro (Brazil), (2016).
19. Zamir A.R. and M. Shah M.: *Accurate image localization based on Google maps Street View*. In: 11th European Conference on Computer Vision, pages 255–268 (2010).
20. Zhang J., and Singh S.: *Visual-lidar Odometry and Mapping: Low drift, Robust, and Fast*. In Proc. of IEEE Int. Conf. on Robotics and Automation (2015).
21. Zhang W. and J. Kosecka J.: *Image based localization in urban environments*. In: Third International Symposium on 3D Data Processing, Visualization, and Transmission, pages 33–40 (2006).