HAL
open science

Improving Robustness of Monocular Urban Localization using Augmented Street View

Li Yu, Cyril Joly, Guillaume Bresson, Fabien Moutarde

To cite this version:

Li Yu, Cyril Joly, Guillaume Bresson, Fabien Moutarde. Improving Robustness of Monocular Urban Localization using Augmented Street View. 19th IEEE International Conference on Intelligent Transportation Systems (ITSC'2016), Nov 2016, Rio de Janeiro, Brazil. hal-01425632

HAL Id: hal-01425632

https://hal.science/hal-01425632v1

Submitted on 3 Jan 2017

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.

Improving Robustness of Monocular Urban Localization using Augmented Street View

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

***Abstract*— With the fast development of Geographic Information Systems, visual global localization has gained a lot of attention due to the low price of a camera and the practical implications. In this paper, we leverage Google Street View and a monocular camera to develop a refined and continuous positioning in urban environments: namely a topological visual place recognition and then a 6 DoF pose estimation by local bundle adjustment. In order to avoid discrete localization problems, augmented Street View data are virtually synthesized to render a smooth and metric localization. We also demonstrate that this approach significantly improves the sub-meter accuracy and the robustness to important viewpoint changes, illumination and occlusion.**

I. INTRODUCTION

Accurate localization is a prerequisite for most autonomous navigation and intelligent driving systems. Recent advances in Geometric Information Systems (GIS), such as Google Street View, Apple's 3D Maps and Bing maps [1], have brought a novel horizon to address the urban localization. GIS become more and more precise and offer a unified global representation of the world with visual, topological, spatial and geographic information [2], [3] . This triggers a boom of visual re-localization systems since it is easy to deploy and affordable to apply. Rather than Simultaneous Localization and Mapping (SLAM), such visual place recognition approaches dispense with a consistant map building and can even render a global position directly when GIS offer geodetic information.

However, the place recognition is still challenging for three major reasons: First, the scene is difficult to identify due to appearance changes caused by viewpoint, illumination, occlusion and structure modifications. Also, an efficient searching method is required to cope with abundant sources in GIS. Third, re-localization is often limited at a topological level by recognizing discrete scenes. A series of methods has been proposed regarding to these three aspects in the literature, like FABMAP [4] and SeqSLAM [5].

In [6], we presented a monocular urban localization system, which pushes the conventional appearance-based localization forward to the metric pose estimation by a graph optimization process. The only input of this approach is an image sequence from a monocular camera and Google Street View. Moreover, it is not necessary to establish frame to frame correspondances, nor the odometry estimates. Although a considerable proportion of localization achieves in a $2m$ accuracy, the discontinuity and drifts still disturb the robustness of the system. We showed that a Street View located far away from the query image ($> 8m$ in the test) generates a significant error in the metric pose estimation phase. The main contribution of this paper is thus an extension of our former framework with the construction of augmented Street Views database that compensates the sparsity of Street Views and improves the localization precision, see in Fig. 1.

Fig. 1. The output from a single localization run using original Street Views and synthesizing virtual views: The trajectories obtained with/without virtual views are plotted in green/blue respectively. The ground truth in red line is recorded by a centimeter-level real-time kinematic GPS (RTK-GPS).

A city-scale Street View database is already too immense to deal with. Therefore, a reasonable way to augmenting virtual views is required and we should also consider how to efficiently render virtual views from the original GIS database, how to encode and index the whole database, and

[1]Authors are with the Centre of Robotics, MINES ParisTech, PSL Research University, 60 Bd Saint Michel, 75006 Paris, France firstname.name@mines-paristech.fr

[2]Authors are with the Institut VEDECOM, 77 rue des Chantiers, 78000 Versailles, France firstname.name@vedecom.fr

how many virtual views to be generated.

To address these issues, we adopt Google Street View as our original database for its world-wide coverage, public accessibility, high resolution geo-referenced panoramas and well-calibrated depth maps. Panoramas and their associated depth maps are used to render virtual views. Instead of synthesizing views in a dense sample grid surrounding candidate panoramas, we only synthesize views between consecutive panoramas according to prior topology information. In this way, obtained virtual views are thus located along the vehicle trajectory with known absolute positions. The coarse topological localization is realized by a Bag-of-Word (BoW) based place recognition algorithm, i.e., the query image, captured by a camera-equipped vehicle, is associated with geotagged Street View images when they share high appearance similarities. Then these Street View candidates as well as their virtual views, are fed to a graph optimization process by a Perpective-n-Point (PnP) algorithm [7]. The vehicle's 6 DoF transformation can be computed from it. As the GPS coordinates of Street Views are known, the vehicle's global localization can be obtained directly. In the experiments, we show that this approach significantly improves the positioning accuracy, continuity and the robustness to important viewpoint changes, illumination and occlusion.

II. RELATED WORK

Visual localization has received a significant attention in the robotics and computer vision community. It can be divided into *topological* methods which realize localization using a collection of images or places; and *metric* methods which estimate a precise pose relative to a map. In the literature, topological localizers focus on robust performances in different scenarios, such as large-scale environments [8], large viewpoint changes [9] and cross-season or illumination variations [3], [10]. Instead, metric localizers work mostly in the fashion of visual odometry (VO) and SLAM [11]. They construct frame-to-frame correspondances and a data fusion scheme to guarantee the continuity of the localization.

Today more and more researchers develop localization systems by using GIS data, such as available maps [12], street network layout [13], geotagged traffic signs [14], satellite images [15] and 3D texture city models [16]. They utilize one of the above sources as a constraint to optimize their localization within a VO or SLAM framework. For instance, Agarwal et al. [17] developed an urban localization with a sub-meter accuracy by modeling a two-step non-linear least squares estimation. They first recover 3D points position from a mono-camera sequence by an optical flow algorithm and then compute the rigid body transformation between Street Views and the estimated points. Alternatively, our coarse-to-fine localizer considers the entirety of topo-metric information in the GIS and construct a metric localization out of the conventional fusion based techniques. The work is inspired by Majdik et al.[16], who leverage Street Views as a geo-referenced image collection to topologically localize a micro aerial vehicle (MAV) and then refine the result by a pose estimation with 3D cadastral models. Considering the query images from a MAV, an important issue is the ground-aerial image matching under big viewpoint changes. But for an urban vehicle localization, query images capture more repetitive scenes along the trajectory which ask for a more robust place recognition algorithm.

As demonstrated in our former work [6], a metric localization without frame-to-frame correspondances introduces a severe discontinuity and drifts. Thus we propose to generate virtual views from original Street View database to address this problem. In general, synthesizing virtual views is mainly used for image matching under extreme viewpoint changes [18], scene registration from a given 3D construction [19], and image descriptor association with laser scans [20]. Torri et al. [10] show that dense sampled virtual views enable true 24/7 place recognition across major illumination changes. In our work, virtual views are rendered by fixing the virtual camera on the subsection between two closest panoramas. In Google Street View, the average distance between consecutive panoramas is around 6 to 16m. Relying on synthesis virtual views, this discrete distance can be reduced within 2 to 5m.

III. METHOD

In this section, we describe our localization algorithm in detail. As illustrated in Fig. 2, the system is divided in two phases: in the offline stage, 4 useful datasets are extracted from Street View, including topology, geo-coordinate, panorama and associated depth map. We pre-process every panorama by rendering rectilinear images from its camera locus and at same time by generating virtual views deviating away from that locus. All rectilinear images are used to build a dictionary by a BoW algorithm. In the online stage, for every query image, we retrieve its most similar rectilinear images (namely referenced images in Fig. 2) from the dictionary. Once referenced images are founded, their neighboring virtual views are also used to construct 3D-to-2D transformation constraints. The global metric localization is obtained by a local bundle adjustment [21] on all the constraints. More technical details are given in the following sections.

A. Preprocessing for Augmented Street View

Google Street View is a planet-scale GIS with billions of panoramic imagery and depth maps (see Fig. 3). They can be publicly accessed by parsing online browsable metadata. Every panorama is stitched by perspective images from a R7 camera system [22]. This system is a rosette of 15 cameras with 5-megapixel CMOS sensors. It enables to register a panorama with a $13,312 \times 6,656$ pixel resolution by capturing a $360°$ horizontal and $180°$ vertical field-of-view (FoV). More importantly, all panoramas are geotagged with accurate GPS coordinates and cover the street in a nearly uniform way. The topological information of a panorama is given by the global yaw angle α, which measures the rotation angle in clockwise direction around the R7 system locus relative to the true north. The associated depth map stores the distance and orientation of various points in the scene coming from

laser range scans or optical flow methods. It only encodes the scene's structural surfaces by their normal directions and distances, allowing to map building façades and roads while ignoring smaller dynamic entities. In fact, the GPS position of Street View is highly precise due to a careful global optimization, while the depth map provides a coarse 3D structure of the scene with a relatively low accuracy. For this reason, we neglect ineffective pixels with a more than $200m$ depth.

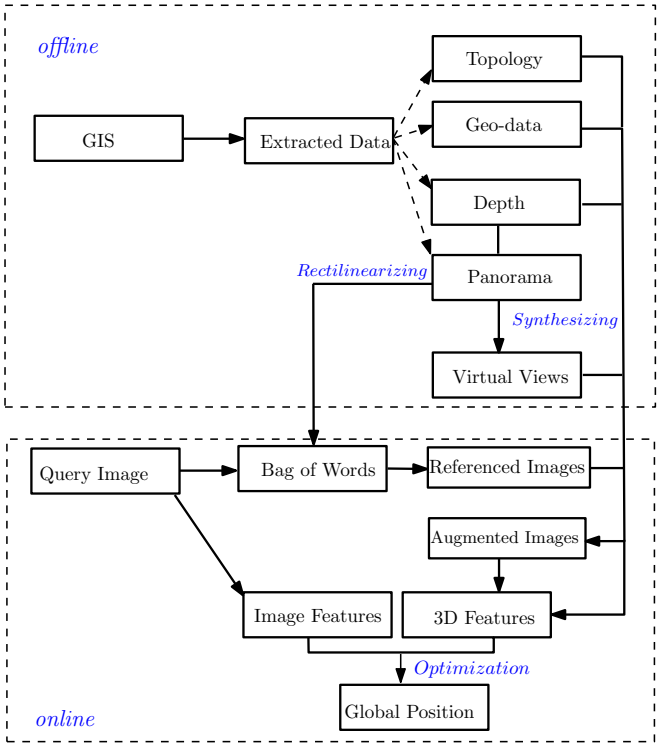

Fig. 2. Flowchart illustrates workflows between different modules.

Panoramas differ from our mono-camera images in both size and visual appearance. To succeed in the metric localization, we carry out two processings: First, panoramas must be transformed into a set of overlapping or unrelated cutouts (rectilinear images) to reduce the large angle distortion. We build a back-projection model by standard ray tracing with bilinear interpolation to realize a more robust and flexible extraction, see details in [6]. We assume 8 virtual pinhole cameras with the camera matrix K are mounted in the centre of a unit sphere $\mathbf{S}$ with a user fixed pitch δ and heading ζ, local yaw angle η changing by $[0°, 45°, ..., 360°]$. The number of virtual cameras, intrinsic matrices and heading/pitch/yaw angles are free to select, yet empirically the more identical they are to the actual on-board camera, the better performance expected. Suppose a 3D point $M \in \mathbb{R}^3$ defined in the sphere coordinates $\mathbf{S}$ at point O, it can be projected directly on a virtual camera image plane $\mathbf{I}(m)$ with a pixel value interpolated from m.

$$m = K\frac{\mathbf{R}_1(\delta,\zeta,\eta)M}{z(\mathbf{R}_1(\delta,\zeta,\eta)M)} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \frac{\mathbf{R}_1(\delta,\zeta,\eta)M}{z(\mathbf{R}_1(\delta,\zeta,\eta)M)} \tag{1}$$

with the focal length f and the principal point(u_0, v_0). We choose the convention that the z axis is the optical axis and normalize rotated 3D points to the unit plan ($z = 1$). These intrinsic parameters are fixed according to our on-board camera. The camera extrinsic matrix is deduced from the above configuration, namely the local heading/pitch/yaw setup. $\mathbf{R}_1$ returns a rotation matrix through the Rodrigues' formula.

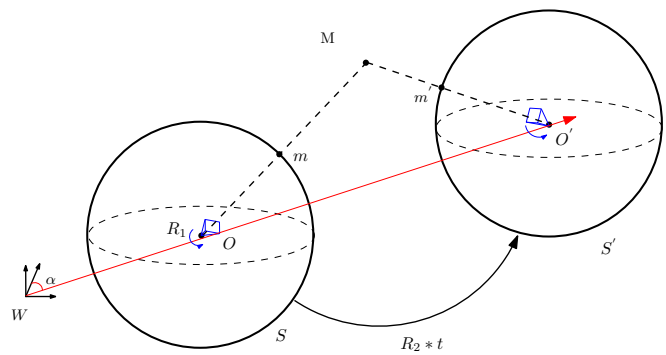

Fig. 4. A virtual panorama at centre point $O^{'}$ is constructed from the original panorama at point O.

Second, instead of fixing virtual cameras in the centre of a unit sphere to get rectilinear images, the camera position of new virtual views is translated along the Google segment corresponding to our trajectory, see Fig. 4. In the test, we generate virtual views from a panorama in both forth and back directions. Inaccurate or missing depth information naturally causes artifacts and absent pixels, for example, sky pixels often disappear. Rendering virtual views from multiple panoramas can potentially improve quality of virtual views. However, in our experiments, we only use a simple synthesizing process since artifacts are usually located in moving objects, such as vehicles or pedestrians, which never appear in our own query images. Our compact preprocessing enables every panorama to have two synthesized virtual panoramas in its neighborhood. Then rectilinear virtual views can be obtained from virtual panoramas via back-projection model. After the artificial generation, the Street View database is augmented with the created views and becomes nearly 10 times larger in quantity.

The rendering pipeline is stated in the following: For every pixel in a virtual panorama, a ray is cast from the centre of a virtual camera and intersect it with the planar 3D structure of its close panorama. The intersection is then projected back to spheric panorama and the depth map is updated according to the transformation. Then, using the back-projection model, we extract rectilinear virtual views from virtual panorama. Pixel values are rendred by bilinear interpolation.

As depicted in Fig. 4, we translate an original panorama $\mathbf{S}$ located at point O to the virtual panorama $\mathbf{S}^{'}$ at point $O^{'}$. The

Fig. 3. An example of Street View panorama (top-left) and its associated depth map (top-right) at location $[48.799133, 2.127752]$ in Versailles, France. The below 8 rectilinear images and depth maps are extracted from the above ones by the back projection. The 8 virtual pinhole cameras are configured similarly to the vehicle's camera, including the same focal length and image size. It creates overlapping views.

translation $\mathbf{t}$ is realized along the trajectory direction with a global yaw angle α. Here, $\mathbf{t}$ is denoted in a east-north-up world coordinates $\mathbf{W}$ by:

$$\mathbf{t} = \begin{bmatrix} l\sin(\alpha) \\ l\cos(\alpha) \\ 1 \end{bmatrix} \tag{2}$$

with l as the distance between two panoramas as $OO^{'}$. Also, the 3D point M is defined in the sphere coordinates at point $\mathbf{S}$ as:

$$M = \begin{bmatrix} d\cos(\theta)\sin(\phi) \\ d\sin(\theta)\sin(\phi) \\ d\cos(\phi) \\ 1 \end{bmatrix} \tag{3}$$

where (θ, ϕ, ρ) is a spherical parametrization and d is the depth information. Therefore, the point M can be projected to $m^{'}$ on a unit sphere $\mathbf{S}^{'}$ by transforming to the coordinates of point $O^{'}$:

$$m^{'} = \frac{M + \mathbf{R}_2(\alpha)\mathbf{t}}{\|M + \mathbf{R}_2(\alpha)\mathbf{t}\|} \tag{4}$$

where $\|M + \mathbf{R}_2(\alpha)\mathbf{t}\|$ is the updated depth $d^{'}$ and $\mathbf{R}_2$ is the rotation matrix deduced from the global yaw angle. Thus $\mathbf{R}_2(\alpha)\mathbf{t}$ describes the relative translation between two panoramas. The pixel values at $m^{'}$ will be as the same as that at m if satisfying $d^{'} > 0$. Then, rectilinear virtual views are registered by the former back-projection model. In order to lessen the influence of absent pixels, we create 12 virtual pinhole cameras to capture more details in virtual views for a good matching. Fig. 5 shows a generation example of two virtual views from a panorama along its topology.

B. Coarse to Fine Localization

For clarity, we briefly review the coarse-to-fine localization system that we proposed in [6]. We feed histogram equalized [23] synthesis views into a BoW training system by combining their SIFT and MSER descriptors. A K-means clustering is used to group useful descriptors as visual words in a dictionary. Next we represent input query images by

Fig. 5. Illustration of virtual views synthesized and back-projected from the original Street View panorama at location $[48.799133, 2.127752]$. The location of the rectilinear Street View, and the $4m$ forward and backward virtual views are colored respectively in red, orange and blue. The black arrows indicate the virtual camera view directions. Missing depth information causes null pixels in the virtual view $\sharp 1$.

histograms of visual words. Term Frequency-Inverse Document Frequency (TF-IDF) reweighing is used to remove redundancy in histograms and an efficient cosine similarity distance metric is computed to retrieve the most similar referenced images *w.r.t* query images. We also explore the inner correlation in the dictionary to speed up the retrieving efficiency.

Then, we construct our metric localization by building constraints existing among an input query image, its most similar referenced image and several Street View images analogous to this referenced image. Constraints are constructed by an inlier 3D-to-2D matching via simple SIFT descriptors. We finally obtain the metric pose and global position in a $g2o$ optimization framework [24]. The 6 DoF pose of the vehicle $\Theta = (\mathbf{R}, t)$, parametrized in Lie algebra $\mathbf{SE}(3)$, is computed by minimizing the reprojection error

under matching constraints:

$$\Theta^{\star} = \arg\min_{\Theta} \sum_{i} \pi \left(\| m_i - \mathbf{P}\left(M_i, \Theta\right) \| \right) \tag{5}$$

where π is a M-estimator based on Tukey Biweight function [25] and $\mathbf{P}\left(M_i, \Theta\right)$ is the image projection from the scene point M_i. The 3D-to-2D correspondence is improved by a RANSAC alogorithm.

In our new method presented here, we still follow the coarse to fine fashion. However, we only establish constraints among an input query image, its most similar referenced image and corresponding synthesized virtual views. It seems that we reduce the number of constraints but in fact closer virtual views can bring stronger constraints to improve the optimization result. In the experiments, we show that the metric location works well even if conventional SIFT/SURF descriptors are used.

IV. Experimental Results and Evaluation

In the test, we evaluated our system on several streets at the city center of Versailles, France. All query images were captured in grayscale by a MIPSee camera, with a $57.6°$ FoV and a 20 frame-per-second frequency, mounted on the vehicle. Only one-side of city façades was captured. The localization ground truth was recorded by a RTK-GPS coupled with a high-accuracy IMU. The intrinsic parameters of virtual cameras and correponding rotation matrices $\mathbf{R}_1(\delta, \zeta, \eta)$ were fixed according to our own MiPSee camera. In order to qualify the metric accuracy, we only selected localization runs when the RTK-GPS reached a below-to-$20cm$ precision. A test example is illustrated in Fig. 1. In this test, our vehicle acquired 1046 query images along a $498m$ trajectory where 28 Google panoramas exist. For every panorama, we generated its virtual views both its forward and backward virtual views.

A. Translation Distance Evaluation

The localization accuracy decreases with the increase of the distance to its referenced Street View. Thus, the translation distance l must be chosen carefully. The main criteria to fix the distance between a panorama and virtual views depends on the following aspects:

- Null pixels and artifacts will be produced if the synthesized views are far away from the rendered panorama. Normally, the farther they are, the more pixels are lost. Consequently, our following metric localization will be influenced significantly.
- We use the global yaw angle to determine of the segment we are traveling on in the topology. Nevertheless, a long translation distance will generate virtual views out of the current street, especially at some narrow crossroads where multiple panoramas meet together. Synthesizing views in such cases is not useful.
- Also, we expect that the final augmented Street View can achieve a uniform distance between every consecutive views' positions. If the translation distance is too small, uniformity can hardly be realized.

We tried several translation distances along the same trajectory in order to find an ideal choice. The geodetic locations were plotted to measure their uniformity. Table I shows the evaluation results. When the translation was fixed to $4m$, as shown in Fig. 1 and Fig. 5, we acquired a compact and uniform distribution of geo-tagged virtual views. After discarding virtual cameras located in buildings, we synthesized 53 virtual panoramas in total. In order to reduce the effects of null pixels existing in virtual views, we fixed 12 virtual cameras to back project virtual panoramas to rectilinear virtual views. Finally, $53 * 12$ synthesized views were added to the Street View database.

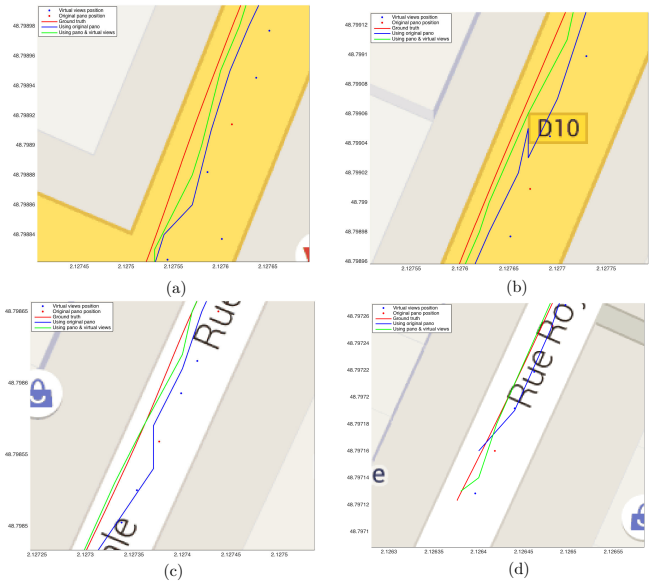

Fig. 6. 4 close-up views of the localization result.

B. Robustness & Accuracy of Localization

Once the augmented database was constructed, the global urban localization could be realized according to our proposed method. Fig. 1 shows an overview of two localization estimated respectively by original and augmented Street Views, and their 4 close-up views from Fig. 1 are provided in Fig. 6. As we proved, the performance of metric localization depends on the distance between the query image and the Street View retrieved in the topological localization. Along the whole trajectory, we only plotted the localization when topology localization reached in a $4m$ accuracy.

Translation distance	$2m$	$4m$	$6m$	$8m$
Invalid camera position	0	3	11	27
Uniform distribution	N	Y	Y	N
Ratio of virtual views with null pixels	0	0.125	0.5	1

TABLE I

Evaluation of the translation distance.

As can be seen from Fig.1, our approach generally works more smoothly and accurately than the localization without virtual views. The path estimated by augmented Street View

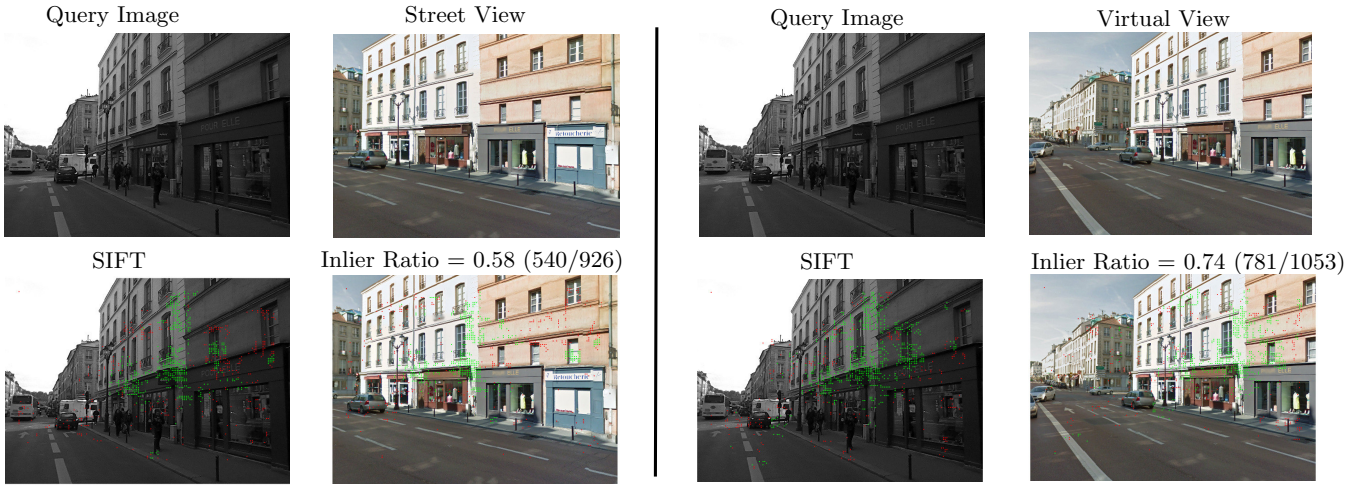

Fig. 7. The same query image is matched with highly similar Street View retrieved by the BoW and with corresponding virtual view. The FLANN based matches are displayed in red and geometrically verified matches are shown in green. The inlier ratio is measured by proportion of geometrically verified matches.

	Original Street View	Augmented Street View
Continuity	$137/1046$	$281/1046$
Average Error	$3.82m$	$3.19m$
Ratio in $[0m, 1m[$	21.89%	41.28%
Ratio in $[1m, 2m[$	28.47%	27.40%
Ratio in $[2m, 3m[$	44.53%	19.22%
Ratio in $[3m, 4m]$	5.11%	12.10%

TABLE II
EVALUATION OF THE LOCALIZATION PERFORMANCE.

is much closer to the ground truth. Virtual Street Views can reduce the accumulative drifts effectively when the vehicle is far away from original panorama, prevent that the localization jumps back while the vehicle is moving forward, make the localization path smoother and longer, see Fig. 6(a) to Fig. 6(d) respectively.

Further, we quantify the performance by using several statistic terms as calculated in Table II. First, we define the continuity as a term to evaluate how many query images can be located within a $4m$ accuracy after the metric localization. We calculated their average error regarding to the ground truth during the whole run. As seen from table, the average accuracy of localization improves a lot and nearly 68.68% of used query images can reach in an error interval $[0m, 2m[$. In contrast, most localization precision stays in $[2m, 4m[$ using original Street View.

Additionally, we analyzed the 3D-2D matches between these localized query images *w.r.t* their Street View and virtual views, see Fig. 7. In the literature, virtual views are often used to improve matching under extreme viewpoint changes. In our former work, we adopted a complex Virtual Line Descriptor (kVLAD) [26] to determine the inlier feature point correspondances when query image was far away from Street View. In the test, we found augmented virtual views can reduce the viewpoints changes and increase inlier matches between query and referenced images as well. After using virtual views, inliers of matches increase significantly. Thus we can simply use SIFT to deal with large viewpoints and illumination changes and reduce the computation cost as well.

V. CONCLUSIONS

We have presented a new method to improve the localization accuracy of a coarse-to-fine approach using Google Street View imagery augmented with synthesized views. Instead of densely sampling the images, we take advantage of the topological information to render useful virtual views in a very sparse way, which enables to lighten the optimization burden and to use simpler descriptors to extract the constraints. These augmented virtual views also allow the system to be more robust to illumination, occlusion and viewpoint changes. We experimentally show that an augmented Street View based monocular localization system works more accurately, smoothly and compactly than direct using the original database.

ACKNOWLEDGMENT

This work is jointly supported by the Institut VEDECOM of France under the autonomous vehicle project and the China Scholarship Council.

REFERENCES

[1] [Online]. Available: https://developers.google.com/maps/
[2] W. Zhang and J. Kosecka, "Image based localization in urban environments," in *3D Data Processing, Visualization, and Transmission, Third International Symposium on*. IEEE, 2006, pp. 33–40.
[3] G. Vaca-Castano, A. R. Zamir, and M. Shah, "City scale geo-spatial trajectory estimation of a moving camera," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 1186–1193.
[4] M. G. Wing, A. Eklund, and L. D. Kellogg, "Consumer-grade global positioning system (gps) accuracy and reliability," *Journal of forestry*, vol. 103, no. 4, pp. 169–173, 2005.
[5] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1643–1649.

[6] L. Yu, C. Joly, G. Bresson, and F. Moutarde, "Monocular urban localization using street view." arXiv preprint arXiv:1605.05157, 2016.

[7] R. C. Bolles and M. A. Fischler, "A ransac-based approach to model fitting and its application to finding cylinders in range data." in *IJCAI*, vol. 1981, 1981, pp. 637–643.

[8] D. M. Chen, G. Baatz, K. Köser, S. S. Tsai, R. Vedantham, T. Pylvä, K. Roimela, X. Chen, J. Bach, M. Pollefeys *et al.*, "City-scale landmark identification on mobile devices," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 737–744.

[9] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 1470–1477.

[10] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1808–1817.

[11] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 55–81, 2015.

[12] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in *Robotics: Science and Systems*, vol. 4. Citeseer, 2007, p. 1.

[13] G. Floros, B. van der Zander, and B. Leibe, "Openstreetslam: Global vehicle localization using openstreetmaps," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1054–1059.

[14] X. Qu, B. Soheilian, and N. Paparoditis, "Vehicle localization using mono-camera and geo-referenced traffic signs," in *Intelligent Vehicles Symposium (IV), 2015 IEEE*. IEEE, 2015, pp. 605–610.

[15] C. U. Dogruer, B. Koku, and M. Dolen, "Global urban localization of outdoor mobile robots using satellite images," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 3927–3932.

[16] A. L. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza, "Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles," *Journal of Field Robotics*, vol. 32, no. 7, pp. 1015–1039, 2015.

[17] P. Agarwal, W. Burgard, and L. Spinello, "Metric localization using google street view," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 3111–3118.

[18] Q. Shan, C. Wu, B. Curless, Y. Furukawa, C. Hernandez, and S. M. Seitz, "Accurate geo-registration by ground-to-aerial image matching," in *3D Vision (3DV), 2014 2nd International Conference on*, vol. 1. IEEE, 2014, pp. 525–532.

[19] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2599–2606.

[20] D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt, "Sift-realistic rendering," in *3D Vision-3DV 2013, 2013 International Conference on*. IEEE, 2013, pp. 56–63.

[21] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd, "Real time localization and 3d reconstruction," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1. IEEE, 2006, pp. 363–370.

[22] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver, "Google street view: Capturing the world at street level," *Computer*, no. 6, pp. 32–38, 2010.

[23] R. A. Hummel, "Histogram modification techniques," *Computer Graphics and Image Processing*, vol. 4, no. 3, pp. 209–224, 1975.

[24] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

[25] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and vision Computing*, vol. 15, no. 1, pp. 59–76, 1997.

[26] Z. Liu and R. Marlet, "Virtual line descriptor and semi-local matching method for reliable feature correspondence," in *British Machine Vision Conference 2012*, 2012, pp. 16–1.