Monocular Urban Localization using Street View

Li Yu*,† Cyril Joly* Guillaume Bresson† Fabien Moutarde*
*Centre of Robotics, MINES ParisTech, PSL
60 Blvd St Michel, 75006, Paris, France
Email: firstname.name@mines-paristech.fr
†Institut VEDECOM
77 Rue des Chantiers, 78000, Versailles, France
Email: firstname.name@vedecom.fr

***Abstract*—This paper presents a metric global localization in the urban environment only with a monocular camera and the Google Street View database. We fully leverage the abundant sources from the Street View and benefits from its topo-metric structure to build a coarse-to-fine positioning, namely a topological place recognition process and then a metric pose estimation by local bundle adjustment. Our method is tested on a 3 km urban environment and demonstrates both sub-meter accuracy and robustness to viewpoint changes, illumination and occlusion. To our knowledge, this is the first work that studies the global urban localization simply with a single camera and Street View.**

I. Introduction

The past decade has seen a substantial progress in the autonomous vehicles field. However, the metric localization in urban environment is still one of crucial challenges [1]. The Global Positioning System (GPS) suffers from a precision degradation and even fails in "urban canyons". The Simultaneous Localization and Mapping (SLAM) requires to build a large consistent map before the localization. The data fusion technique based on GPS, proprioceptive sensors and existing maps, is still tricky to evaluate the confidence of different data sources and its localization quality depends a lot on sensors' price.

Alternatively, Geographical Information Systems (GIS), such as Google Street View, Mappy, Navteq, etc., are more and more precise and offer a unified global representation of our world with visual, topological, spatial and geographic information [2]. This motivates us to fully take advantage of the abundant sources in GIS to realize a metric localization in urban contexts. In this paper, Google Street View is adopted for its word-wide coverage, public accessibility, high resolution geo-referenced panoramas and well-calibrated depth [3]. We develop a coarse-to-fine position estimation system. At the coarse level, a topological localization is addressed by a place recognition algorithm. Namely, a query image captured by our camera equipped on vehicle can be associated with geotagged Street View images when sharing high appearance similarities. The vehicle's position is thus restricted on an intersected area around the geodetic position of corresponding Street View cameras. At this stage, an efficient retrieval method is described to shrink the search time. Then, since the depth maps are integrated in Street View, we can refine the before-mentioned location by computing the vehicle's 6 degrees of freedom (DoF) transformation *w.r.t* Street View images by solving the Perspective-n-Point (PnP) problem [4]. Finally, a local Bundle Adjustment (LBA) [5] is applied to improve the consistency of the localization regarding to the multiple Street View images retrieved.

In fact, extensive efforts have been made to solve the urban localization by using GIS data such as available maps [6], street network layout [7], geotagged traffic signs [8], satellite images [9] and 3D texture city models [10]. They focus on one of the above sources as a single strong constraint to optimize their tasks and rarely consider the entire topo-metric information in the GIS, for instance, Street View is commonly employed only as an image database input to solve the place recognition problem [11]. Numerous papers pay attention to the localization in a large-scale environments, with the view point change or under cross-season or light variation [12], [13]. Their pipeline is to construct a Bag-of-Words (BoW) model [14], query an input image within the model and localize the place by comparing images vectors' similarity. On the other hand, in our approach, we propose to consider the place recognition problem as a primary selection to feed the metric localization. Moreover, our approach consider the multiple database constraints to speed up the searching instead of a pure statistic BoW matching.

In a similar fashion to our work, Agarwal et al. [15] realize an urban localization with a sub-meter accuracy by modeling a two-phase non-linear least squares estimation. They first recover 3D points position from a mono-camera sequence via an optical flow and then compute the rigid body transformation between the Street Views and the estimated points. Street Views are learnt as a BoW model and retrieve efficiency is improved via an inaccurate GPS input to narrow the searching radius. Their key contribution is to localize globally on Google Maps without a heavy map construction. This method still follows a SLAM manner. A map building is omitted but numerous 3D scene points must be estimated and registered as usual. The Street View imagery serves as keyframes to solve the loop closing problem. Rather, our method fully depends on the Street View and the only input is just a stream of mono-camera images without odometry or GPS information.

II. Method

In this section, we describe the coarse-to-fine localization system in detail. A pseudo-code is provided in Algorithm 1 to give an overview. Note that the system is divided into two phases. The Street View panoramas are prepared and their rectilinear views are generated offline (Line 1 to 7) prior to an actual online vehicle localization (Line 8 to 18). In the offline part, all generated imagery is trained as a BoW database. A symmetric comparison matrix is computed for the database so as to reduce the time to retrieve close images in online phase. It also serves as a index reference for the following optimization steps. In the online stage, for every vehicle query image, we extract top similar images from the database as the topological localization and then estimate relative poses between each other. Finally, a LBA is employed with all estimated poses and corresponding 3D-to-2D matching constraints to obtain the global metric localization.

Algorithm 1 Metric global localization in the urban area

```
Input: Street View panoramas $\mathbf{S} = \{S_1, S_2, \dots, S_n\}$ and
       their depth maps $\mathbf{D} = \{D_1, D_2, \dots, D_n\}$
Input: Query images $I_q = \{I_1, I_2, \dots, I_m\}$ captured by a
       vehicle driving in the city
Output: Global location of the vehicle
 1: for $i \leftarrow 1$ to $n$ do
 2:     Rectilinear processing of S and D
 3:     Feature extraction and BoW training
 4:     Construction of $DB$ as the final BoW dictionary
 5: end for
 6: Calculation of the intra-distance matrix $D_{DB\times DB}$ and
    speed-up matrix $D_{DB\times r}$        ▷ cf. Section II-B
 7: Up to here the algorithm is implemented offline.
 8: for $t \leftarrow 1$ to $n$ do
 9:     Parametrize $I_q$ by BoW
10:     Search $I_q$'s the best similar image $I_r$ in $DB$
11:     Get the top $k$ similar database images of $I_r$ in $D_{DB\times r}$
12:     Histogram Equalization: $I'_{t/r} = E\left(I_{t/r}\right)$
13:     for $j \leftarrow 1$ to $k$ do
14:         Feature matching
15:         Pose estimation $\Theta_j$        ▷ cf. Equation 4
16:     end for
17:     $\Theta^{\star} = LBA\left\{\Theta_j\right\}$
18:     Recover the global position at step $t$
19: end for
```

A. Data preparation and generation

Google Street View is a rich street-level GIS with millions of panoramic imagery and depth maps captured all over the world (see Figure 1) [3]. Every panorama, geotagged with an accurate GPS position, is registered a $13,312 \times 6,656$ pixel resolution by capturing a 360° horizontal and 180° vertical field-of-view (FoV). It is composed and projected by rectilinear images from several cameras. Generally, there is a panorama every 6 to 15m with a nearly uniform street coverage. The associated depth map stores the distance and orientation of various points in the scene via laser range scans or using image motion methods (optical flow). It only encodes the scene's dominant surfaces by its normal direction and its distance, allowing to map building façades and roads while ignoring smaller entities such as vehicles or pedestrians. For the sake of bandwidth saving, it is sampled down to 512×256 pixels but recovering the similar size to the panorama is easy. The GPS position of Street Views is highly precise due to a careful global optimization, while the depth map provides a 3D structure of the scene with a relatively low accuracy [3]. Depending on the route planner built in Street View, all panoramic images and depth maps can be downloaded iteratively. The panoramas ought to be transformed into a set of overlapping or unrelated cutouts to reduce the large angle distortion. Google offers a public API [16] to extract rectilinear images by setting a virtual camera's parameters, including FoV, pitch and yaw. Currently, most of relative researches rely on this API to generate perspective images. Instead, we build a back-projection model to realize a more robust and flexible extraction, as depicted in Figure 2. 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 direction and the following yaw directions $[0°, 45°, ..., 360°]$. The number of virtual cameras, intrinsic matrix and pitch/yaw degree are free to select, yet empirically the more identical they are to the actual on-board camera, the better performance expected. That is one of the reasons why we develop our own rectilinear model. Consider a 3D point $\mathbf{M} \in \mathbb{R}^3$ in homogeneous coordinates using spherical parametrization (θ, ϕ, ρ), denoted as:

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

Its projection on the unit sphere is represent by $\mathbf{m}^*$

$$\mathbf{m}^* = \frac{\mathbf{M}}{\|\mathbf{M}\|} = \begin{bmatrix} \cos(\theta)\sin(\phi) \\ \sin(\theta)\cos(\phi) \\ \cos(\phi) \\ 1 \end{bmatrix} \tag{2}$$

and its projection on a virtual camera image plan by $\mathbf{m}$.

$$\mathbf{m} = K\left[\mathbf{R}, \mathbf{t}\right]\mathbf{P} \tag{3}$$

where the camera extrinsic matrix $[\mathbf{R}, \mathbf{t}]$ is deduced from the above configuration. The perspective virtual views can be synthesized by the rays tracing back to the panorama with a bilinear interpolation around the back-projected points. The warp process is noted as ω.

$$\mathbf{m} \xrightarrow{\omega} \mathbf{m}^* \tag{4}$$

The corresponding perspective depth map can be addressed likewise. A generation example from a Street View panorama of Versailles in France is illustrated in Figure 1.

In the experiment, we have totally collected a test set of 3655 query images captured by a MiPSee camera on the

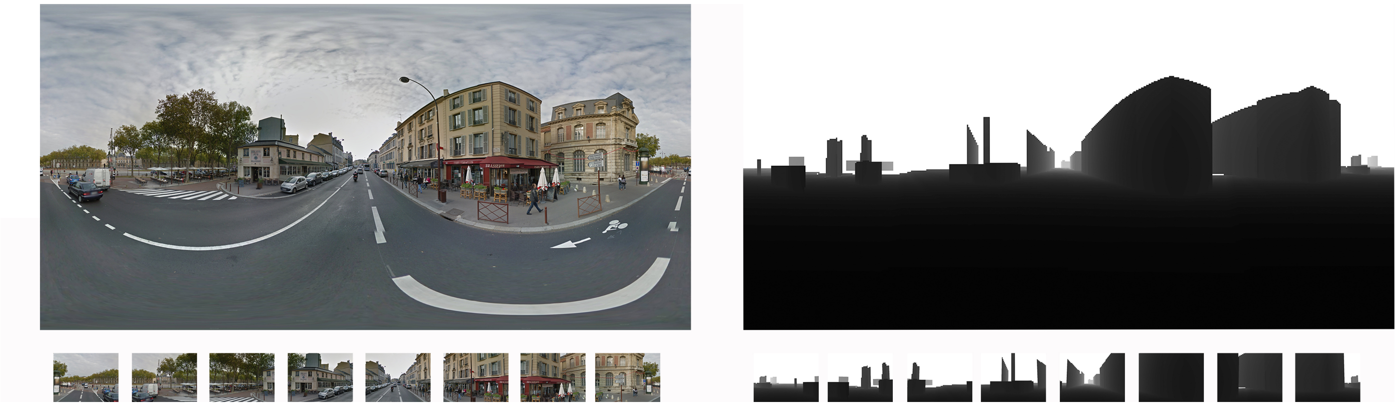


Fig. 1. An example of Street View panorama (top-left) and its associated depth map (top-right) in Versailles, France. The below 8 rectilinear images and depth maps are extracted from the above ones by the back projection. The 8 virtual pinhole cameras, are configured similarly to the vehicle's camera, including the same focal length and image size. It creates overlapping views.

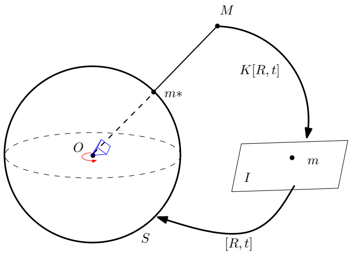


Fig. 2. Back-projection model: 8 virtual cameras are constructed at point **O** and pixels in the image plane **I** are bilinearly interpolated from panorama sphere. The yaw offset changes according to the red arrow direction.

vehicle with a $57.6°$ FoV. Images were captured at the city centre of Versailles. The ground truth at each localization was recorded by a centimeter-level real-time kinematic GPS (RTK-GPS).

B. Database construction and retrieval acceleration

After the artificial generation, the Street View database becomes 8 times larger in quantity. We use BoW retrieval techniques to represent database images as numerical vectors quantized by feature descriptors, and to perform a hierarchical clustering (K-means) of the image descriptors in a tree structure. After the weighing strategies, the whole tree is referred to as a dictionary and its leaves as visual words. A topological localization is estimated according to a distance criterion based on the vectors similarity. In our test, we use the Term Frequency-Inverse Document Frequency (TF-IDF) reweighing and the efficient cosine similarity distance metric. In particular, considering the dynamic changes in viewpoints, illumination and occlusions, we construct two independent dictionaries generated from the SIFT and MSER detectors, and then normalize them. The final dictionary can take into account both local and regional feature descriptors.

As a rule of thumb, the bigger the database is, the slower will be the information retrieval from it. The aim now is to facilitate the run-time search even if a metropolis database is constructed. In a natural manner, we explore the intra database similarities and integrate the potential topological information to reduce the computational cost. Figure 3 shows the similarity between images within the whole database $D_{DB\times DB}$ as well as the relationship between the database and query images $D_{DB\times Q}$. The results, detailed in the figure caption, show a certain regularity to locate top similar images *w.r.t* a query image. To some degree, these regularities reflect the topology and inner connections in the GIS, e.g. all database images are downloaded successively according to the Street View route planner. Intuitively, neighboring panoramas share similar appearances and correspondingly their similar rectilinear images ought to situate in an analogous yaw or auxiliary angle of yaw offset (only for two-way streets). That is the reason why many yellow parallel lines and vertical dark lines lie in $D_{DB\times DB}$ and $D_{DB\times Q}$ respectively. All these regularities make our database a sparse "searching map", which are advantageous to significantly reduce the number of image comparisons in the online phase. It is summarized as follows:

- In a piecewise route database $D_{DB\times DB}$, for each database image I_r we register its top k similar database images' $I_{0\ldots k}$ location in an array $D_{DB\times r}$.
- A maximum appearance distance $dist_{max}$ is defined as the farthest radius around a panorama where its neighboring panoramas share an overlapping view. Normally, 6 to 8 nearest panoramas share a common scene around 50 to 70 meters. In $D_{DB\times DB}$, the $dist_{max}$ can be fixed as 48 or 64 matrix index steps.
- With the help of the route planner, we can localize the first incoming query image easily by searching the database image around the starting point. Then for an ordinary query image, its searching range can be narrowed to all $D_{DB\times r}$ within the threshold $dist_{max}$.

According to our experiment, the above approach can avoid nearly 90% of the comparisons without any retrieval loss compared to a whole database search.

C. Global metric localization

In the previous sections, a candidate set of images has been retrieved from database. In order to realize a metric lo-

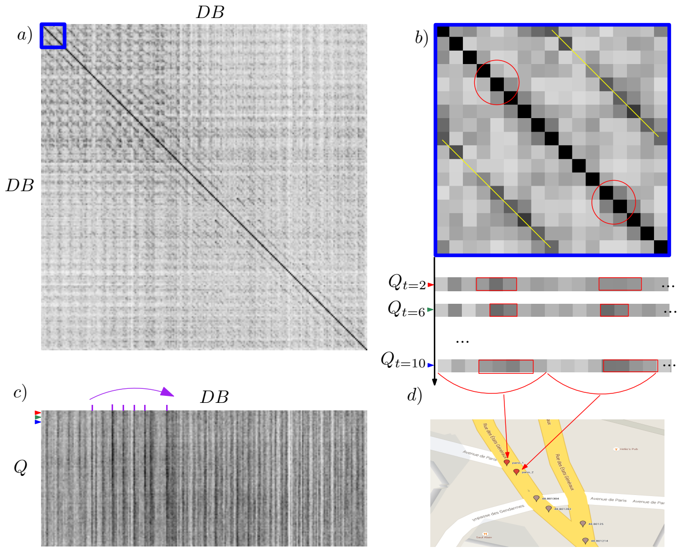


Fig. 3. Illustration of distance matrices and intra-relations: *a*) The intra-database symmetric matrix $D_{DB\times DB}$. The rows and columns represent 240 rectilinear views generated from 30 panoramas in a 350m urban route, and the matrix intensity is computed by mapping the cosine similarity from $[0, 1]$ to $[0, 255]$, therein, the darker pixels depict the higher similarities and diagonal values are always equal to 255. *b*) The above matrix is a close-up view *w.r.t* the blue rectangle in *a*), it shows two obvious yellow lines with high intensity, that are parallel to the diagonal line indicating a location regularity for similar images in the matrix. *c*) The distance matrix $D_{DB\times Q}$ rows represent the 240 database images and columns represent a query sequence with 100 frames. Several darkest vertical lines are highlighted by purple marks, meaning that a short query sequence can find its most similar database images only in few range of the database as shown by the purple arrow. Considering the candidate location at query time steps $t = 2, 6, 10$, we compare their close-ups to the database shown in *b*) and conclude that even in the same panorama, the similar candidates sparsely focus on some certain yaw offsets. The red cycles and rectangles represent top similar candidates from the same panorama. *d*) The panoramas are searched at 3 time steps in *b*).

calization, high-inlier feature correspondence between camera images and the candidate set should be guaranteed. Other than a statistic comparison with the BoW, the conventional perspective matching is done by following this kind of pipeline [17]: (a) a set of keypoints are extracted in both query and database images, e.g. SIFT; (b) their descriptors are matched by nearest neighbor search algorithms, e.g. FLANN; (c) matching outliers are rejected by the ratio test and geometric verification using constraints from the homograhy or the fundamental matrix, e.g. RANSAC 8-point algorithm. Unfortunately, we found this pipeline is unstable and inaccurate because of important changes from viewpoint, illumination and occlusion in urban area.

To counter this effect, we use the histogram equalization [18] to enhance the contrast of all images prior to feature extraction. Also the Virtual Line Descriptor (kVLD) [19] is applied to determine the inlier feature point correspondences. It is a SIFT-like descriptor by signing virtual lines to the points with geometrical consistency. The algorithm computes and matches a k connected virtual line graph to reject the outliers, see Figure 6.

Figure 4 depicts our global metric localization process. In this system, the vehicle captures images at consecutive camera states x_{t-1} and x_t. No odometric input is integrated between successive states. At the state x_t, the best database image r_1 associated to the query image is retrieved through the topological localization. As mentioned before, via the database $D_{DB\times r}$, the top $k-1$ analogous database images to the best one are found and denoted as $[r_2, r_3, \ldots, r_k]$. Naturally, we suppose that the query image shares the overlapping view with these $k-1$ images as well. The constraints between them are found thanks to the accurate matching features. With the help of depth maps in the database, the 6 DoF pose of the vehicle $\Theta = (\mathbf{R}, \mathbf{t})$, parametrized in Lie algebra $\mathbf{SE}(3)$, can be computed by minimizing the reprojection error between a pair of images, i.e.:

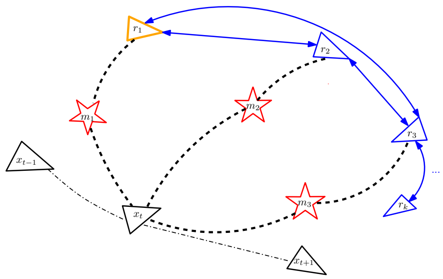


Fig. 4. Illustration of Local Bundle Adjustment to estimate the global position of the vehicle. The vehicle, the best similar database image and other top $k-1$ similar database images are the triangles respectively colored in black, orange and blue. The red stars represent good matching features between the query and database images.

$$\Theta^{\star} = \arg\min_{\Theta} \sum_i \pi\left(\|\mathbf{m}_i - \mathbf{P}\left(\mathbf{M}_i, \Theta\right)\|\right) \tag{5}$$

where $\mathbf{P}(\mathbf{M}_i, \Theta)$ is the image projection from the scene point $\mathbf{M}_i$. The 3D-to-2D correspondence is improved by RANSAC and π, as a M-estimator based on Tukey Biweight function [20], is used to improve convergence and to cope with wrong correspondence.

$$\pi(x) = \begin{cases} t^2 \,/\, 6 \left(1 - \left[1 - \left(\frac{x}{t}\right)^2\right]^3\right) & \text{if } |x| \leq t \\ t^2 \,/\, 6 & \text{if } |x| > t \end{cases} \tag{6}$$

Finally, k poses and corresponding image pairs are put into a LBA to refine the vehicle's pose and its global position.

We use the optimization framework $g2o$ [21] as our non-linear least squares solver. The error regarding to the feature detection is assumed to follow a Gaussian distribution. The k geotagged views extracted from panoramas are constrained in Figure 4 and their camera configuration are given. We can use them to verify the performance of our metric localization since their absolute positions are known. Moreover, the parameters of the LBA, can be first initialized by this verification as well.

III. Experimental Evaluation and Discussion

In order to evaluate the performance of our algorithm, we test on several city streets and record query images when the RTK-GPS reaches a localization within 20cm precision. Using this kind of RTK-GPS positions as the ground truth is sufficient for our metric accuracy qualification. In the experiment, our vehicle is mainly tested on the roads around the area with GPS coordinates $[48.801631, 2.131509]$. The intrinsic parameters of Google virtual cameras are fixed according to our own MiPSee camera. Only one-side of city façades are captured.

The performance of our coarse-to-fine algorithm depends closely on both topological and metric localization part. In fact, if the coarse topological localization fails, the further metric part based on it works in vain. However, in city scenario, it is difficult to construct a strict ground truth to evaluate the topological localization. In [10] and [22], authors proposed a manual labeling way to build up a confusion matrix as the ground truth. This method is too time consuming to employ as our test dataset contains more than 3000 frames. As expected, the visual overlap must exist between the query and database image if the BoW algorithm works well. Therefore we can evaluate the topological localization based on the geometric consistency. If the inlier-match number between two images is larger than 12, we can get a bound with a 19.8m radius around the Street View in which the corresponding query images lie, as shown by the red circle in Figure 5. The bound radius distance is calculated between the RTK-GPS of query images and the geo-data of retrieved Street View. After manually verifying the pairs matches along a street with 33 panoramic Street Views, our method is able to recognize 100% visual similar database images corresponding to query images. The 264 rectilinear Street Views are trained to a mixed 5000 SIFT and 2000 MSER visual words.

Based on the previous topological localization, we choose the EPnP-RANSAC to compute the pose between each query monocular image and matched Street View. The solution is selected with a highest consensus by setting a minimum reprojection error within 3 pixels and maximum inliers. Then a $g2o$ framework is used to optimize all computed solutions. This process is realized under the cartesian coordinates with the Lambert conformal conic projection and the result is converted back to geo-coordinates under the WSG84.

Figure 5 shows the metric global localization in a $287m$ city street where 423 query images and 13 panoramic Street Views are used. There are 58 metric positions obtained within an error of 6.5m *w.r.t* the RTK-GPS and 58.6% keeps within a 2m accuracy. An example of matches between the query and retrieved Street View is depicted in Figure 6.

Although the accuracy is higher than the regular autonomous GPS (average error of 8m) [23], readers may still wonder the discontinuity (only $58/423$ frames localized) and drifts. The cause of errors during the metric localization thus is analyzed in two aspects: the topological distance between the chosen Street View and RTK-GPS ground truth, and the inner matches between the query and Street View image. The inlier-match number reflects the feature quantity that is used in the 3D-to-2D pose estimation process, and the real topological distance between the query and Street View image can describe the feature quality. Commonly, the less or the further the features are tracked, the less accurate the localization is. We select all query images that retrieve the same

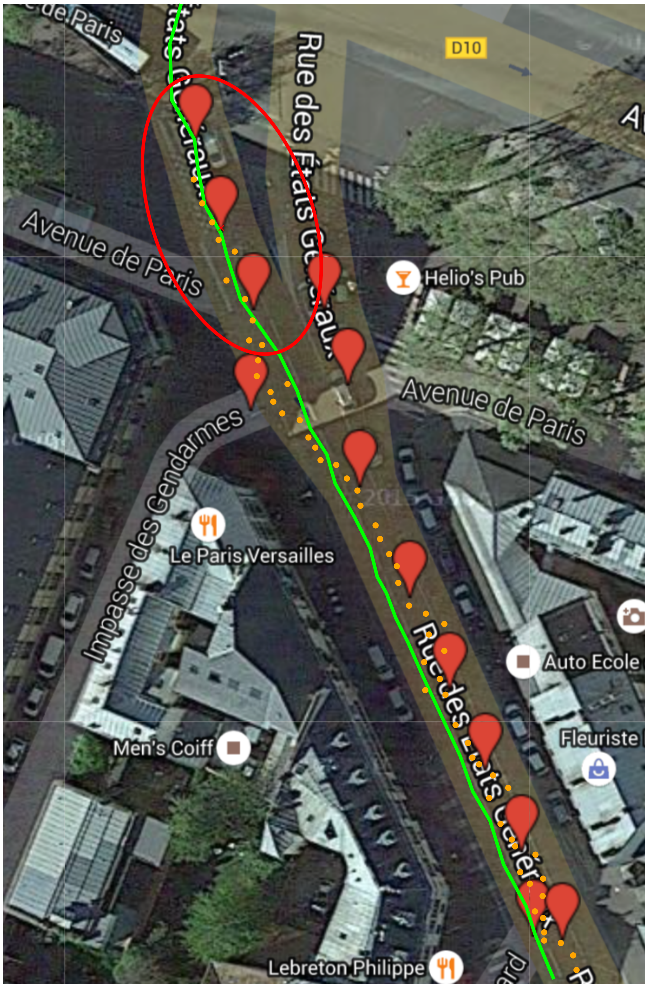
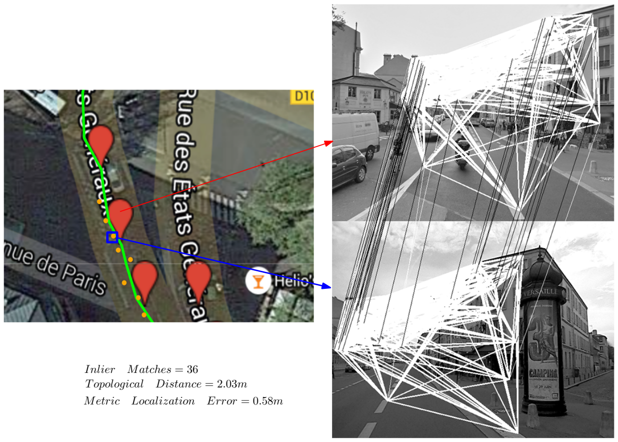


Fig. 5. Bird's-eye view of the metric global localization. The red Google pointers represent the locations of the Street View cameras. The orange dots mark the estimated positions of the monocular camera. The green line illustrates RTK-GPS ground truth and the red circle shows the topological bound around a panorama denoted by the cross mark.



D10
Rue des Etats Gé
ue de Paris
Helio
Inlier Matches = 36
Topological Distance = 2.03m
Metric Localization Error = 0.58m

Fig. 6. An example of matches between a monocular image and its retrieved Street View image by kVLAD descriptors. The metric localization reaches a 0.58m accuracy by refining from the topological localization (2.03m).

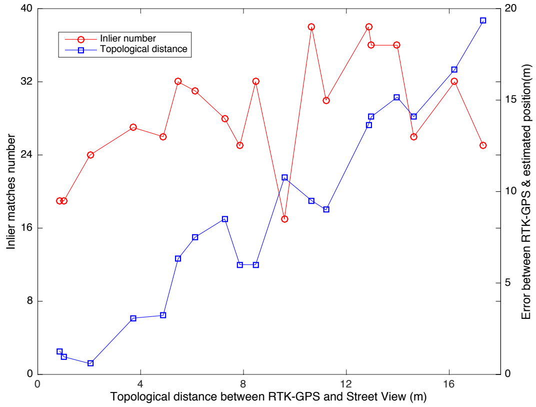


Fig. 7. Metric error analysis *w.r.t* inlier-match numbers and the topological distance.

Street View located at the coordinates $[48.801631, 2.131509]$ and obtain the relationships in Figure 7. Our effective BoW algorithm guarantees the visual overlaps between query images and retrieved Street Views around 20m bound. In this way, the inlier-match number affects little to the drifts. However, the errors appear in a cumulative trend when topological distance increases between query and Street View. Despite the optimization improves the accuracy, too many further features only show small motions and still causes disastrous errors even passing the bound of topological localization. As a result, we eliminate lots of localization like this and cause unavoidable localization discontinuity.

Naturally the synthesis view from the Street View can be created to augment nearer features and enhance the accuracy. This will be one of our future work.

IV. Conclusion

To conclude, this paper presents a metric urban localization by matching 3D Street View points to a monocular camera. Our approach requires neither the construction of a consistent map nor the prior visit of the environment. This simple set-up makes our algorithm affordable and easy to be widely deployed in real urban localization. In order to realize a reliable localization, we throughly explores multiple information of Street View and strictly respects their intra relationships. Our technique demonstrates both the high accuracy *w.r.t* the Street View and robustness in complicated urban environments.

Acknowledgment

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

References

[1] 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.

[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] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] [Online]. Available: https://developers.google.com/maps/

[17] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

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

[19] 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.

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

[21] 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.

[22] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza, "Mav urban localization from google street view data," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3979–3986.

[23] 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.