X-View: Graph-Based Semantic Multi-View Localization

Abel Gawel*, Carlo Del Don*, Roland Siegwart, Juan Nieto and Cesar Cadena

***Abstract*—Global localization in multi-view robot data is a challenging task. Appearance-based global localization approaches often fail under drastic view-point changes, as representations have limited view-point invariance. This work is based on the idea that human-made environments contain rich semantics which can be used to disambiguate global localization. Here, we present *X-View*, a Multi-View Semantic Global Localization system. *X-View* leverages semantic graph descriptor matching for global localization, enabling localization under drastically different view-points. While the approach is general in terms of the semantic input data, we present and evaluate an implementation on visual data. We demonstrate the system in experiments on the publicly available *SYNTHIA* dataset, on a realistic urban dataset recorded with a simulator, and on real-world StreetView data. Our findings show that *X-View* is able to globally localize aerial-to-ground, and ground-to-ground robot data of drastically different view-points. Our approach achieves an accuracy of up to $85\,\%$ on global localizations in the multi-view case, while the benchmarked baseline appearance-based methods reach up to $75\,\%$.**

***Index Terms*—Localization, Semantic Scene Understanding, Mapping**

I. INTRODUCTION

GLOBAL localization between heterogeneous robots is a difficult problem for classic place-recognition approaches. Visual appearance-based approaches such as [1, 2] are currently among the most effective methods for re-localization. However, they tend to significantly degrade with appearance changes due to different time, weather, season, and also view-point [3, 4]. In addition, when using different sensor modalities, the key-point extraction becomes an issue as they are generated from different physical and geometrical properties, for instance intensity gradients in images vs. high-curvature regions in point clouds.

Relying on geometrical information, directly from the measurements or from a reconstruction algorithm, on the other hand shows stronger robustness on view-point changes, seasonal changes, and different sensor modalities. However, geometrical approaches typically do not scale well to very large environments, and it remains questionable if very strong view-point changes can be compensated while maintaining only a limited overlap between the localization query and database [5, 6].

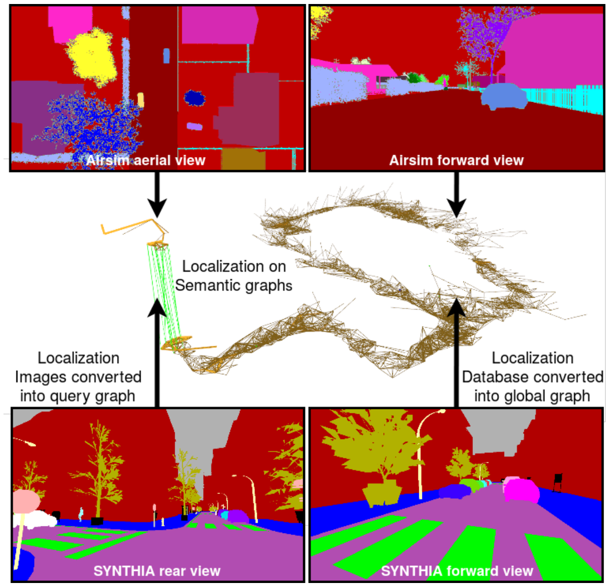

Airsim aerial view
Airsim forward view
Localization on Semantic graphs
Localization Images converted into query graph
Localization Database converted into global graph
SYNTHIA rear view
SYNTHIA forward view

Figure 1: *X-View* globally localizes data of drastically different view-points using graph representations of semantic information. Here, samples of the experimental data is shown, i.e., semantically segmented images from the publicly available *SYNTHIA* and the *Airsim* datasets. The localization target graph is built from data of one view-point (*right images*), while the query graph is built from sequences of another view-point (*left images*). *X-View* efficiently localizes the query graph in the target graph.

Another avenue to address appearance and view-point changes are Convolutional Neural Network (CNN) architectures for place recognition [4, 7]. While these methods show strong performance under appearance changes, their performance is still to be investigated under extreme view-point variations.

Recently, topological approaches to global localization regained interest as a way to efficiently encode relations between multiple local visual features [8, 9]. On the other hand, the computer vision community has made great progress in semantic segmentation and classification, resulting in capable tools for extracting semantics from visual and depth data [10–12].

Based on the hypothesis that semantics can help to mitigate the effects of appearance changes, we present *X-View*, a novel approach for global localization based on building graphs of semantics. *X-View* introduces graph descriptors that efficiently represent unique topologies of semantic objects. These can be matched in much lower computational effort, therefore not suffering under the need for exhaustive sub-graph matching [13].

This work was supported by European Union's Seventh Framework Programme for research, technological development and demonstration under the TRADR project No. FP7-ICT-609763 and by the National Center of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation.

* The authors contributed equally to this work.

Authors are with the Autonomous Systems Lab, ETH Zurich. gawela@ethz.ch, deldonc@student.ethz.ch, {rsiegwart, nietoj, cesarc}@ethz.ch

By using semantics as an abstraction between robot view-points, we achieve invariances to strong view-point changes, outperforming CNN-based techniques on RGB data. Furthermore, with semantics understanding of the scene, unwanted elements, such as moving objects can naturally be excluded from the localization. We evaluate our global localization algorithm on publicly available datasets of real and simulated urban outdoor environments, and report our findings on localizing under strong view-point changes. Specifically, this paper presents the following contributions:

- A novel graph representation for semantic topologies.
- Introduction of a graph descriptor based on random walks that can be efficiently matched with established matching methods.
- A full pipeline to process semantically segmented images into global localizations.
- Open source implementation of the *X-View* algorithm[1].
- Experimental evaluation on publicly available datasets.

The remainder of this paper is structured as follows: Sec. II reviews the related work on global localization, followed by the presentation of the *X-View* system in Sec. III. We present our experimental evaluation in Sec. IV and conclude our findings in Sec. V.

II. Related Work

In this section we review the current state-of-the-art in multi-robot global localization in relation to our proposed system.

A common approach to global localization is visual feature matching. A large amount of approaches have been proposed in the last decade, giving reliable performance under perceptually similar conditions [1–3]. Several extensions have been proposed to overcome perceptually difficult situations, such as seasonal changes [14, 15], daytime changes [4, 16], or varying view-points using CNN landmarks [7, 17]. However, drastic view-point invariance, e.g., between views of aerial and ground robots continues to be a challenging problem for appearance-based techniques.

In our previous work, we demonstrated effective *3D* heterogeneous map merging approaches between different view-points from camera and LiDAR data, based on overlapping *3D* structural descriptors [5, 6]. However, 3D reconstructions are still strongly view-point dependent. While these techniques do not rely on specific semantic information of the scenes, the scaling to large environments has not yet been investigated, and computational time is outside real-time performance with large maps.

Other approaches to global localization are based on topological mapping [18, 19]. Here, maps are represented as graphs $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$ of unique vertices $\boldsymbol{V}$ and edges $\boldsymbol{E}$ encoding relationships between vertices. While these works focus on graph merging by exhaustive vertex matching on small graphs, they do not consider graph extraction from sensory data or ambiguous vertices. Furthermore, the computationally expensive matching does not scale to larger graph comparisons.

With the recent advances in learning-based semantic extraction methods, using semantics for localization is a promising avenue [20–22]. In [21, 22] the authors focus on the *data association* problem for semantic localization using Expectation Maximization (EM) and the formulation of the pose estimation problem for semantic constraints as an error minimization. The semantic extraction is based on a standard object detector from visual key-points.

Stumm et al. [8] propose to use graph kernels for place recognition on visual key-point descriptors. Graph kernels are used to project image-wise covisibility graphs into a feature space. The authors show that graph descriptions can help localization performance as to efficiently cluster multiple descriptors meaningfully. However, the use of large densely connected graphs sets limitations to the choice of graph representation. Motivated, by these findings, we propose to use graph descriptors on sparse semantic graphs for global localization.

III. X-View

In this section, we present our Graph-Based Multi-View Semantic Global Localization system, coined *X-View*. It leverages graph extraction from semantic input data and graph matching using graph descriptors. Fig. 2 illustrates the architecture of the proposed global localization algorithm, focusing on the graph representation and matching of query semantic input data to a global graph. The localization target map is represented as the global graph. *X-View* is designed to operate on any given odometry estimation system and semantic input cue. However, for the sake of clarity, we present our system as implemented for semantically segmented images, but it is not limited to it.

A. System input

We use semantically segmented images containing pixel-wise semantic classification as input to the localization algorithm. These segmentations can be achieved using a semantic segmentation method, such as [11, 12]. Also instance-wise segmentation, i.e., unique identifiers for separating overlapping objects of same class in the image space can be considered for improved segmentation, but is not strictly necessary for the approach to work. Furthermore, we assume the estimate of an external odometry system. Finally, we also consider a database semantic graph $\boldsymbol{G}_{db}$, as it could have been built and described on a previous run of our graph building algorithm as presented in the next sub-sections.

B. Graph extraction and assembly

In this step, we convert a sequence of semantic images $\boldsymbol{I}_q$ into a query graph $\boldsymbol{G}_q$. We extract blobs of connected regions, i.e., regions of the same class label l_j in each image. Since semantically segmented images often show noisy partitioning of the observed scene (holes, disconnected edges and invalid labels on edges), we smooth them by dilating and eroding the boundaries of each blob. We furthermore reject blobs smaller than a minimum pixel count to be included in the graph, to

[1] https://github.com/ethz-asl/x-view

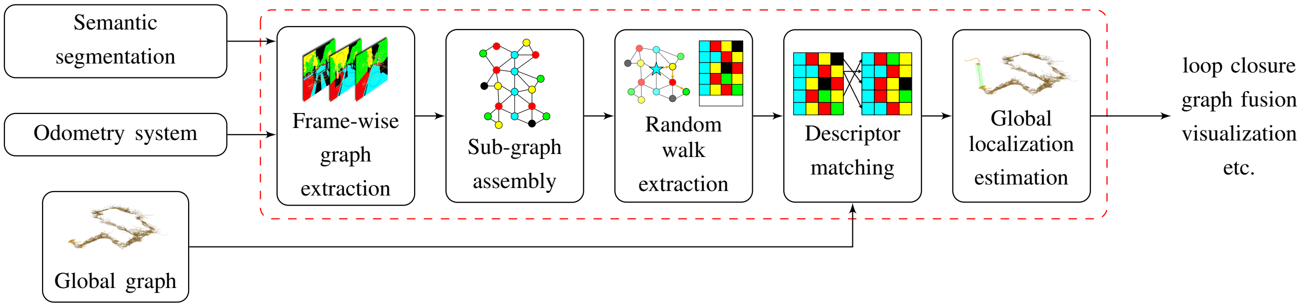


Figure 2: *X-View* global localization system overview. The inputs to the system are semantically segmented frames (e.g., from RGB images) and the global graph $\boldsymbol{G}_{db}$. First, a local graph is extracted from the new segmentation. Then, the sub-graph $\boldsymbol{G}_q$ is assembled and random walk descriptors are computed on each node of $\boldsymbol{G}_q$. The system matches the sub-graph random walk descriptors to $\boldsymbol{G}_{db}$, e.g., recorded from a different view-point. Finally, the matches are transferred to the localization back-end module to estimate the relative localization between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$. Consecutively, the relative localization can be used for various purposes such as loop closure, fusing $\boldsymbol{G}_q$ into $\boldsymbol{G}_{db}$ or for visualization.

mitigate the effect of minor segments. This process removes unwanted noise in the semantically segmented images. The magnitude of this operation is 4 pixels, and has a minor effect on the segmentation result. However, it ensures clean boundaries between semantic segments. Furthermore, the center location $\boldsymbol{p}_j$ of the blobs are extracted and stored alongside the blob labels as vertices $\mathbf{v}_j = \{\boldsymbol{l}_j, \boldsymbol{p}_j\}$. In the case that also instance-wise segmentation is available, it can be considered in the blob extraction step, otherwise the extraction operates only on a class basis.

The undirected edges $\boldsymbol{e}_j$ between vertices are formed when fulfilling a proximity requirement, which can be either in image- or *3D*-space. In the case of image-space, we assume images to be in a temporal sequence to grow graphs over several frames of input data. However, this is not required in the *3D* case.

Using a depth channel or the depth estimation from, e.g., optical flow, the neighborhood can be formed in *3D*-space, using the *3D* locations of the image blobs to compute a Euclidean distance. The process is illustrated for image data in Fig. 3 (top). Then, several image-wise graphs are merged into $\boldsymbol{G}_q$ by connecting vertices of consecutive images using their Euclidean distance, see Fig. 3. To prevent duplicate vertices of the same semantic instance, close instances in $\boldsymbol{G}_q$ are merged into a single vertex, at the location of the vertices' first observation. The strategy of merging vertices into their first observation location is further motivated by the structure of *continuous* semantic entities, such as streets. This strategy leads to evenly spaced creation of *continuous* entities' vertices in $\boldsymbol{G}_q$.

C. Descriptors

X-View is based on the idea that semantic graphs hold high descriptive power, and that localizing a sub-graph in a database graph can yield good localization results. However, since sub-graph matching is an NP-complete problem [13], a different regime is required to perform the graph localization under real-time constraints, i.e., in the order of seconds for typical robotic applications. In this work, we extract random walk descriptors for every node of the graph [23], and match them in a subsequent step. This has the advantage that the descriptors can be extracted and matched in constant or linear time, given a static or growing database-graph, respectively.

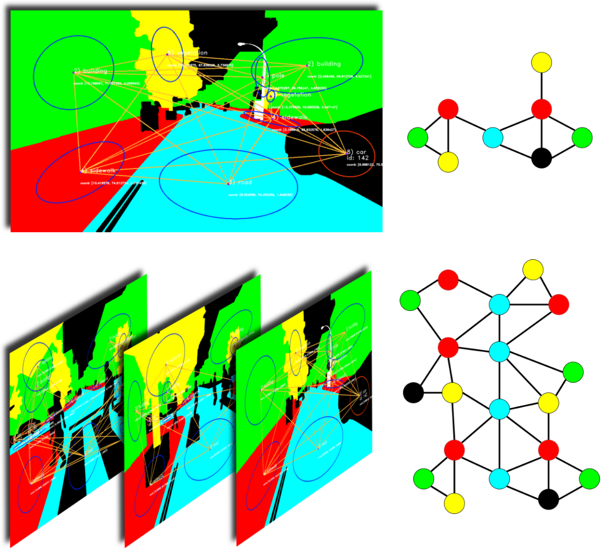


Figure 3: Extraction of semantic graphs from one image (top) and a sequence of images (bottom). Vertices are merged and connected from sequences of input data. Note that we omitted some vertices and edges in the sample graphs on the right side for visualization purposes and reduced the graph to a planar visualization, whereas the semantic graphs in our system are connected in *3D*-space. The ellipses around each vertex were added for visualization and represent a scaled fitted ellipse on a semantic instance of the segmentation image.

Each vertex descriptor is an $n \times m$ matrix consisting of n random walks of depth m. Each of the random walks originates at the base vertex $\mathbf{v}_j$ and stores the class labels of the visited vertices. Walk strategies, such as preventing from immediate returns to the vertex that was visited in the last step, and exclusion of duplicate random walks can be applied to facilitate expressiveness of the random walk descriptors. The process of random walk descriptor extraction is illustrated in Fig. 4.

D. Descriptor Matching

After both $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$ are created, we find associations between vertices in the query graph and the ones in the database graph by computing a similarity score between the corresponding graph descriptors. The similarity measure is computed by matching each row of the semantic descriptor of the query vertex to the descriptor of the database vertex. The number of identical random walks on the two descriptors

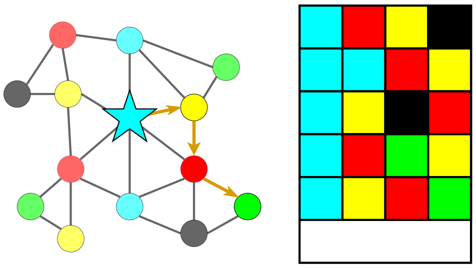


Figure 4: Schematic representation of the random walk extraction. (Left) From a seed vertex, cyan star, the random walker explores its neighborhood. This results in the descriptor of n random walks of depth m (here, $m = 4$). The highlighted path corresponds to the last line of the descriptor on the right. (Right) Each line of the descriptor starts with the seed vertex label and continues with the class labels of the visited vertices.

reflects the similarity score s, which is normalized between 0 and 1. In a second step, the k matches with highest similarity score are selected for estimating the location of the query graph inside the database map.

E. Localization Back-End

The matching between query graph and global graph, the robot-to-vertex observations, and the robot odometry measurements result in constraints $\boldsymbol{\theta}_i \subseteq \boldsymbol{\Theta}(\boldsymbol{p}_i, \boldsymbol{c}_i)$ on the vertex positions $\boldsymbol{p}_i$ and robot poses $\boldsymbol{c}_i$ with $\boldsymbol{\theta}_i = \boldsymbol{e}_i^T \boldsymbol{\Omega}_i \boldsymbol{e}_i$, the measurement errors $\boldsymbol{e}_i$, and associated information matrix $\boldsymbol{\Omega}_i$. Specifically, these three types of constraints are denoted as $\boldsymbol{\Theta}_{\mathbf{M}}(\boldsymbol{p}_i), \boldsymbol{\Theta}_{\mathbf{V}}(\boldsymbol{p}_i, \boldsymbol{c}_i)$, and $\boldsymbol{\Theta}_O(\boldsymbol{c}_i)$ respectively. The matching constraints $\boldsymbol{\Theta}_{\mathbf{M}}(\boldsymbol{p}_i)$ stem from the semantic descriptor matching of the previous step. These associations are added as identity transformations between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$. The robot odometry constraints $\boldsymbol{\Theta}_O(\boldsymbol{c}_i)$ are created using the robots estimated odometry between consecutive robot poses associated to the localization graph. The robot-to-vertex constraints encode the transformation between each robot-to-vertex observation, i.e., $\boldsymbol{c}_i$ and the projected *3D* blob locations $\boldsymbol{p}_i$. Using these constraints, we compute a Maximum a Posteriori (MAP) estimate of the robot pose $\boldsymbol{c}_i$ by minimizing a negative log-posterior $\mathbf{E} = \sum \boldsymbol{\theta}_i$, i.e.,

$$\boldsymbol{c}_i^* = \underset{\boldsymbol{c}_i}{\operatorname{argmin}} \sum \boldsymbol{\Theta}(\boldsymbol{p}_i, \boldsymbol{c}_i) \tag{1}$$

with $\boldsymbol{\Theta}(\boldsymbol{p}_i, \boldsymbol{c}_i) = \{\boldsymbol{\Theta}_M(\boldsymbol{p}_i), \boldsymbol{\Theta}_V(\boldsymbol{p}_i, \boldsymbol{c}_i), \boldsymbol{\Theta}_V(\boldsymbol{p}_i)\}$ This optimization is carried out by a non-linear Gauss-Newton optimizer. Optionally, the algorithm also allows to reject matching constraints either in a sample consensus manner, using RANSAC or using a robust estimator [**?**] on all constraints between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$, excluding the specific constraints from the optimization objective. We initialize the robot position at the mean location of all matching vertices' locations from $\boldsymbol{G}_{db}$.

IV. Experiments

We evaluate our approach on two different synthetic outdoor datasets with forward to rear view, and forward to aerial view, and one real world outdoor dataset with forward to rear view. In this section, we present the experimental set-up, the results, and a discussion.

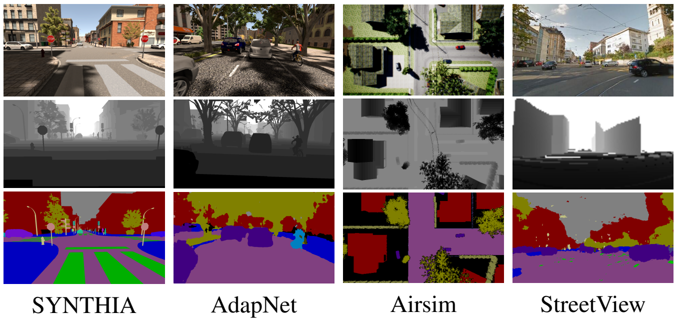


Figure 5: Sample images from the datasets used in the experiments: (top) RGB image, (middle) Depth image, (bottom) Semantic segmentation. (left) *SYNTHIA* with perfect semantic segmentation, (middle left) *SYNTHIA* with *AdapNet* semantic segmentation, (middle right) *Airsim* with perfect semantic segmentation, (right) *StreetView* with *SegNet* semantic segmentation.

A. Datasets

The first of the used datasets is the public *SYNTHIA* dataset [24]. It consists of several sequences of simulated sensor data from a car travelling in different dynamic environments and under varying conditions, e.g., weather and daytime. The sensor data provides RGB, depth and pixel-wise semantic classification for 8 cameras, with always 2 cameras facing forward, left, backwards and right respectively. The segmentation provides 13 different semantic classes which are labelled class-wise. Additionally, dynamic objects, such as pedestrians and cars are also labelled instance-wise. We use sequence 4, which features a town-like environment. The total travelled distance is $970\,m$.

In the absence of suitable public aerial-ground semantic localization datasets, we use the photo-realistic *Airsim* framework [25] to generate a simulated rural environment[2]. This environment is explored with a top-down viewing Unmanned Aerial Vehicle (UAV) and a car traversing the streets with forward-facing sensors. Both views provide RGB, depth and pixel-wise semantic classification data in 13 different classes with instance-wise labelling. Furthermore, both trajectories are overlapping with only an offset in $\mathbf{z}$-direction and have a length of $500\,m$ each. Please note that we used a pre-built environment, i.e., the objects in the environment have not specifically been placed for enhanced performance.

Finally, we evaluate the system on a dataset gathered from *Google StreetView* imagery. The RGB and depth data of a straight $750\,m$ stretch of Weinbergstrasse in Zurich are extracted via the *Google Maps API*[3]. Analogously to the *SYNTHIA* dataset, we use forward and backward facing camera views.

While the travelled distance between two image locations in the *Airsim* dataset is always $1\,m$, it varies between $0\,m$ to $1\,m$ in the *SYNTHIA* dataset, and is approximately $10\,m$ between two frames in the *StreetView* dataset. Sample images of all datasets are depicted in Fig. 5.

Our approach relies on semantic representations of scenes. While we do not propose contributions on semantic extraction from raw sensor data, recent advances on semantic segmentation show ever increasing accuracies on visual and depth

[2] http://robotics.ethz.ch/~asl-datasets/x-view/

[3] https://goo.gl/iBniJ9

data [10–12, 26]. We therefore evaluate the performance on *SYNTHIA* both using semantic segmentation with *AdapNet* [11], and the ground truth as provided by the dataset. On the *Airsim* data, we only use the segmentation from the dataset, and on the *StreetView* dataset, we use semantic segmentation with *SegNet* [12].

B. Experimental Setup

We evaluate the core components of *X-View* in different experimental settings. In all experiments, we evaluate *X-View* on overlapping trajectories and the provided depth and segmentation images of the data. First, we focus our evaluation of the different graph settings on the *SYNTHIA* dataset. We then perform a comparative analysis on *SYNTHIA*, *Airsim*, and *StreetView*.

In *SYNTHIA*, we use the left forward camera for building a database map and then use the left backward camera for localization. Furthermore, we use 8 semantic classes of *SYNTHIA*: *building*, *street*, *sidewalk*, *fence*, *vegetation*, *pole*, *car*, and *sign*, and reject the remaining four classes: *sky*, *pedestrian*, *cyclist*, *lanemarking*. The *AdapNet* semantic segmentation model is trained on other sequences of the *SYNTHIA* dataset with different seasons and weather conditions.

Analogously, we use the forward-view of the car in the *Airsim* dataset to build the database map and then localize the UAV based on a downward-looking camera. Here we use 6 classes *(street, building, car, fence, hedge, tree)* and reject the remaining from insertion into the graph *(powerline, pool, sign, wall, bench, rock)*, as these are usually only visible by one of the robots, or their scale is too small to be reliably detected from the aerial robot.

Finally, in the *StreetView* data, we use the forward view to build the database and localize using the rear facing view. Out of the 12 classes that we extract using the pre-trained *SegNet* model[4], we use five, i.e., *(road, sidewalk, vegetation, fence, car)*, and reject the remaining as these are either dynamic *(pedestrian, cyclist)*, unreliably segmented *(pole, road sign, road marking)*, or omni-present in the dataset *(building, sky)*.

We build the graphs from consecutive frames in all experiments, and use the *3D* information to connect and merge vertices and edges, as described in III-B. The difference between graph construction in image- and *3D*-space is evaluated in a separate experiment. No assumptions are made on the prior alignment between the data. The ground-truth alignment is solely used for performance evaluation.

C. Localization performance

We generate the PR of the localization based on two thresholds. The localization threshold t_L is applied on the distance between the estimated robot position $\boldsymbol{c}_i^*$ and the ground truth position $\boldsymbol{c}_{gt}$. It is set as *true*, if the distance between $\boldsymbol{c}_i^*$ and $\boldsymbol{c}_{gt}$ is smaller than t_L, i.e., $\|\boldsymbol{c}_i^* - \boldsymbol{c}_{gt}\| \leq t_L$, and to *false* for $\|\boldsymbol{c}_i^* - \boldsymbol{c}_{gt}\| > t_L$. The margin t_L on the locations is required, since $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$ do not create vertices in the exact same spot. The same node can be off by up to twice the distance that we use for merging vertices in a graph. Here, we use $t_L = 20\,m$ for *SYNTHIA* and *StreetView*, and $t_L = 30\,m$ for *Airsim*. For the PR curves, we vary the consistency threshold t_c that is applied on the RANSAC-based rejection, i.e., the acceptable deviation from the consensus transformation between query and database graph vertices. The localization estimation yields a positive vote for an estimated consensus value s of $s \leq t_c$ and a negative vote otherwise.

Firstly, we evaluate the effect of different options on the description and matching using the random walk descriptors (i.e., random walk parameters, graph coarseness, number of query frames, dynamics classes, graph edge construction technique, and seasonal changes) as described in Sec. III-B - III-D. To illustrate the contrast to appearance-based methods, we also present results on two visual place recognition techniques based on BoW, as implemented by Gálvez-López and Tardos [2], and NetVLAD [4] on the datasets' RGB data. To generate the PR of the reference techniques, we vary a threshold on the inverse similarity score for BoW, and a threshold on the matching residuals of NetVLAD.

Furthermore, we show the performance of the full global localization algorithm on the operating point taken from the PR curves. Our performance metric is defined as the percentage of correct localizations over the Euclidean distance between $\boldsymbol{c}_i^*$ and $\boldsymbol{c}_{gt}$. As for BoW and NetVLAD, we take localization as the best matching image. The localization error is then computed as the Euclidean distance between associated positions of the matched image and the ground truth image. To improve performance of the appearance-based methods, we select the operating points with high performances, i.e., high precisions in the PR curves.

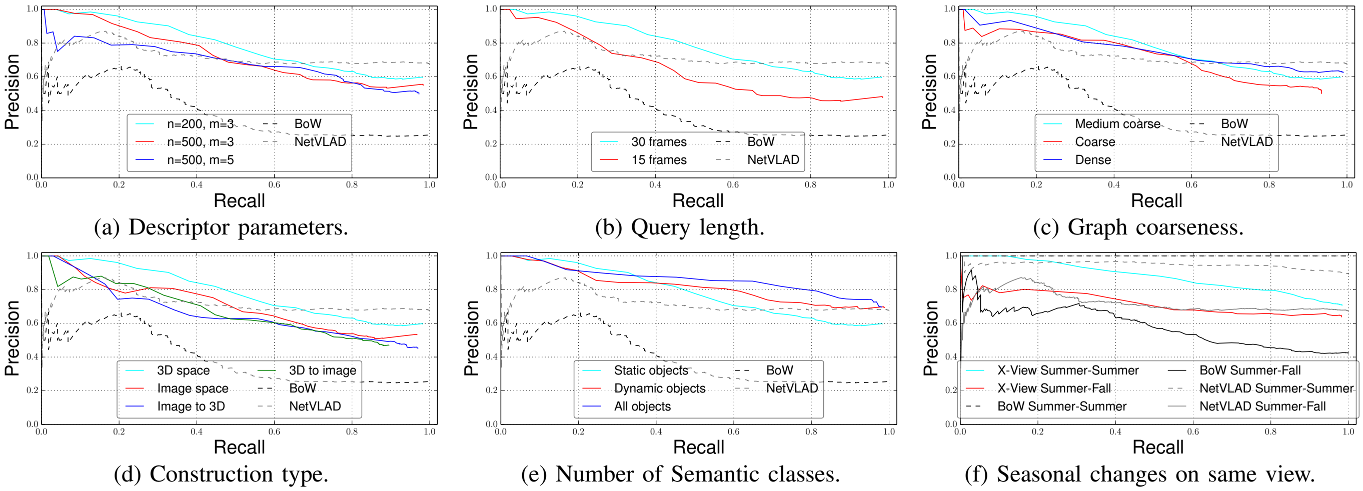
D. Results

While we illustrate the effects of different attributes of *X-View* in Fig. 6 as evaluated on *SYNTHIA*, we then also show a comparison on all datasets in Fig. 7.

Fig. 6a depicts the effect of varying the random walk descriptors on the graph. Here, a descriptor size with number of random walks $n = 200$ and walk depth m between $3 - 5$, depending on the size of $\boldsymbol{G}_q$ perform best. Both decreasing n or increasing m leads to a decrease in performance. These findings are expected, considering query graph sizes ranging between $20 - 40$ vertices. Under these conditions, the graph can be well explored with the above settings. Descriptors with larger walk depth m significantly diverge between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$, as the random walk reaches the size limits of $\boldsymbol{G}_q$ and continues exploring already visited vertices, while it is possible to continue exploring $\boldsymbol{G}_{db}$ to greater depth.

Secondly, Fig. 6b presents PR-curves for different sizes of $\boldsymbol{G}_q$, i.e., different numbers of frames used for the construction of $\boldsymbol{G}_q$. An increase in the query graph size leads to a considerable increase of the localization performance. Also this effect is expected as $\boldsymbol{G}_q$ contains more vertices, forming more unique descriptors. However, it is also desirable to keep the size of $\boldsymbol{G}_q$ limited, as a growing query graph size requires larger overlap between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$. Furthermore, the computational time for descriptor calculation and matching grows with increased query graph size.

[4] goo.gl/EyReyn



(a) Descriptor parameters.

(b) Query length.

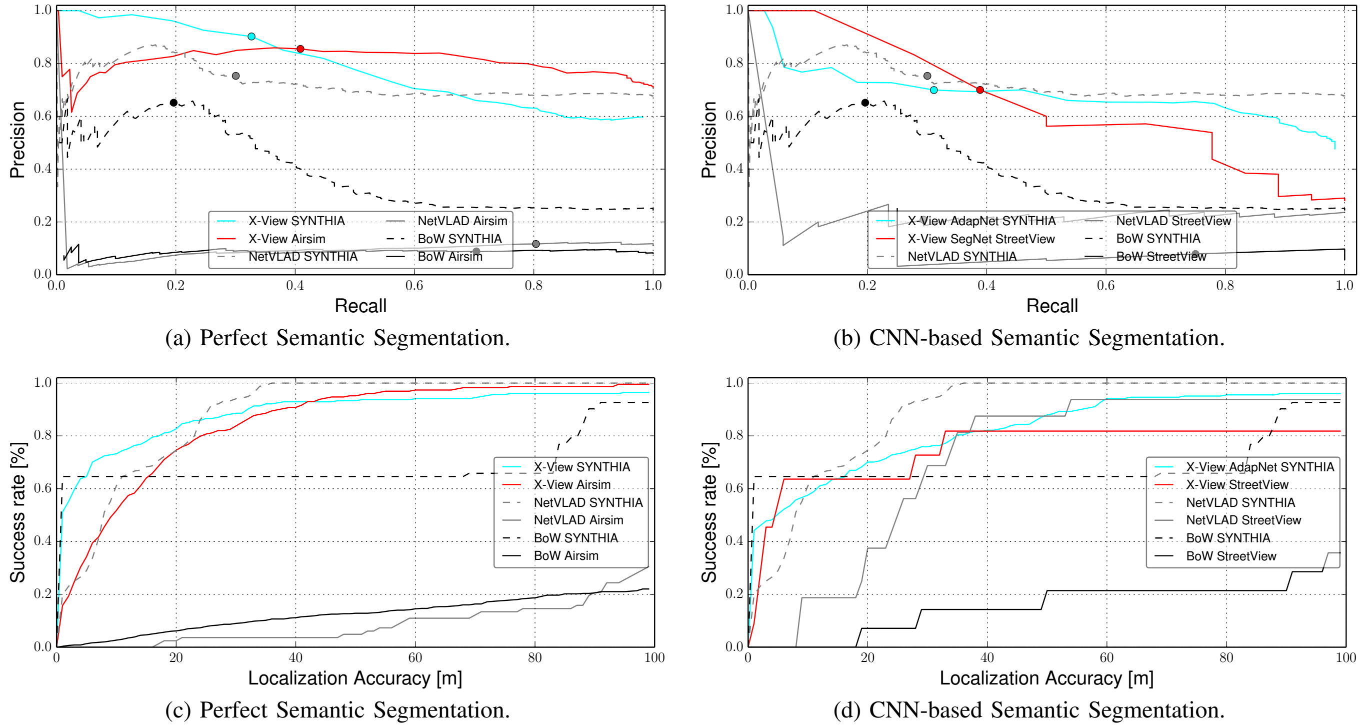
(c) Graph coarseness.

(d) Construction type.

(e) Number of Semantic classes.

(f) Seasonal changes on same view.

Figure 6: PR curves for localization of the rear view semantic images against a database graph built from the forward view on the *SYNTHIA* dataset (except (f)). For all plots we accept a localization if it falls within a distance of $20\,m$ from the ground-truth robot position. This threshold corresponds to the value up to which query graph vertices of the same semantic instance can be off from their corresponding location in the database graph, caused by the graph construction technique. (a) illustrates the effect of different descriptor settings on the localization performance. (b) shows the effect of increasing the amount of frames used for query graph construction, while (c) depicts the effect of using coarser graphs, i.e., a large distance in which we merge vertices of same class label. In (d) we compare the extraction methods in image-, and *3D*-space and in (e) the effect of including all semantic objects against including a subset of semantic classes. Lastly, in (f), we evaluate the localization performance on a configuration with the right frontal camera as query and the left frontal camera for the database, under the effect of seasonal changes. In contrast to the other plots where we use the ground truth, we use semantic segmentation with *AdapNet* on the data. The appearance-based techniques used are visual BoW [2] and NetVLAD [4].



(a) Perfect Semantic Segmentation.

(b) CNN-based Semantic Segmentation.

(c) Perfect Semantic Segmentation.

(d) CNN-based Semantic Segmentation.

Figure 7: Localization performance of *X-View* on the *SYNTHIA*, *Airsim*, and the *StreetView* data compared to the appearance-based methods [2, 4]. The operation points are chosen according to the respective PR curves in (a) and (b), indicated as dots. (c) illustrates the performance on perfectly semantically segmented data on *SYNTHIA*, and *Airsim*. (d) shows the system's performance on the *SYNTHIA*, and *StreetView* datasets using CNN-based semantic segmentation.

Thirdly, Fig. 6c shows the impact of increased graph coarseness, i.e., larger distances of merging vertices. Here, the coarseness cannot be arbitrarily scaled to low or high values, as it leads to either over- or under-segmented graphs. Our best performing results were obtained with a vertex merging distance of $10\,m$ for the *SYNTHIA* dataset, and $15\,m$ for *Airsim* and *StreetView* datasets, respectively.

Fourthly, Fig. 6d illustrates the effect of graph extraction in either image- or *3D*-space. The extraction in *3D*-space, taking advantage of the depth information as described in Sec. III-B shows superior performance. However, *X-View* still performs well when localizing a graph built in one space against a graph built in the other.

Fifthly, Fig. 6e explores the inclusion of different object classes. The configurations are: Only static object classes, static object classes plus dynamic object classes, and all object

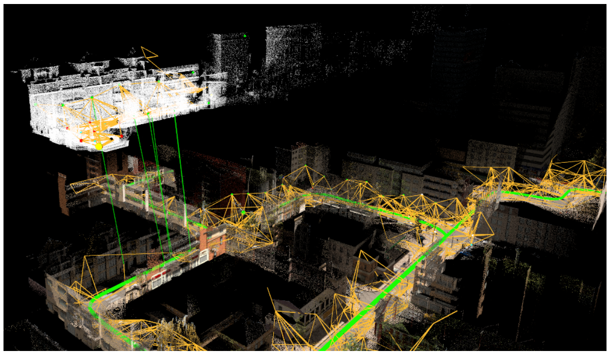


Figure 8: Exemplary localization on the SYNTHIA dataset. The query graph is depicted on the top of the image (embedded in white point-cloud), while the global graph is illustrated on the bottom (embedded in the colored point-cloud). The resulting localizations between the two graphs are depicted as vertical green lines. Please note that the point-clouds are only added for visualization purposes, but are not required for the algorithm.

classes. Here, the results are not conclusive on the *SYNTHIA* dataset and more evaluations will be needed in the future.

Lastly, Fig. 6f shows *X-View*'s performance under seasonal change. We compare the performance of localizing the query graph built from the right forward facing camera of one season in the database graph built from the left forward facing camera of another season. Here, we consider the summer and fall sequences of *SYNTHIA*. The BoW-based techniques perform well in this scenario if the seasonal conditions are equal. However, its performance drastically drops for inter-season localization, while *X-View*, and NetVLAD suffer much less under the seasonal change.

The evaluation using PR-curves, and success rates over the localization error is depicted in Fig. 7. *X-View* has higher success rate in multi-view experiments than the appearance-based techniques on both synthetic datasets at our achievable accuracy of $20\,m$ for *SYNTHIA* and $30\,m$ on *Airsim* and using perfect semantic segmentation inputs as depicted in Fig. 7c. These accuracies are considered successful as node locations between $\boldsymbol{G}_q$ and $\boldsymbol{G}_{db}$ can differ by twice the merging distance with our current graph merging strategy. On the considered operation point of the PR curve, *X-View* achieves a localization accuracy of $85\,\%$ within $30\,m$ on *Airsim*, and $85\,\%$ on *SYNTHIA* within $20\,m$. A qualitative example of the localization is depicted in Fig. 8.

Furthermore, *X-View* expresses comparable or better performance for multi-view localization than the appearance-based techniques using CNN-based semantic segmentation on the *SYNTHIA*, and *StreetView* datasets respectively. Here we consider successful localizations within $20\,m$ for both datasets. The achieved accuracies on the chosen operation points are $70\,\%$ on *SYNTHIA*, and $65\,\%$ on *StreetView*.

Finally, we also report timings of the individual components of our system in Table I. Here, the construction of $\boldsymbol{G}_q$ has by far the largest contribution, due to iteratively matching and merging frames into $\boldsymbol{G}_q$. As the graphs in *SYNTHIA* consider more classes and smaller merging distances, these generally contain more vertices and therefore longer computational times.

Module	*SYNTHIA*	*Airsim*
Blob extraction	2.73 ± 0.65	1.76 ± 0.26
Construction of $\boldsymbol{G}_q$	337.39 ± 92.81	257.40 ± 28.30
Random Walks Generation	1.38 ± 0.82	1.07 ± 0.56
Matching $\boldsymbol{G}_q$ to $\boldsymbol{G}_{db}$	7.30 ± 4.51	4.33 ± 1.25
Localization Back-End	22.50 ± 9.71	5.15 ± 0.63
Total	371.3 ± 108.5	269.71 ± 31.0

Table I: Timing results in ms, reporting the means and standard deviations per frame on the best performing configurations on *SYNTHIA* and *Airsim*. The timings were computed on a single core of an Intel Xeon E3-1226 CPU @ 3.30GHz.

E. Discussion

Global localization with multi-view data is a difficult problem where traditional appearance based techniques fail. Semantic graph representations can provide significantly better localization performance under these difficult perceptual conditions. We furthermore give insights how different parameters, choices, and inputs' qualities affect the system's performance. Our results obtained with *X-View* show a better localization performance than appearance-based methods, such as BoW and NetVLAD.

During our experiments, we observed that some of the parameters are dependent on each other. Intuitively, the coarseness of the graph has an effect on the random walk descriptors as a coarser graph contains fewer vertices and therefore deeper random walks show decreasing performance as $\boldsymbol{G}_q$ can be explored with short random walks. On the other hand, an increasing amount of frames used for localization has the reverse effect on the descriptor depth as $\boldsymbol{G}_q$ potentially contains more vertices, and deeper random walks do not show a performance drop as they do for smaller query graphs.

Also the success rate curves indicate that *X-View* outperforms the appearance based methods particularly in the presence of strong view-point changes. While the appearance-based methods fail to produce interesting results for the *Airsim* dataset, they have a moderate to good amount of successful localizations on *SYNTHIA* and *StreetView*. On the other hand, *X-View* has generally higher localization performance and does not show a strong drop in performance among datasets. While computational efficiency has not been the main focus of our research, the achieved timings are close to the typical requirements for robotic applications.

Finally, we performed experiments both using ground truth semantic segmentation inputs, and CNN-based semantic segmentation. The performance with semantic segmentation using *AdapNet* [11] shows to be close to the achievable performance with ground truth segmentation on *SYNTHIA*. Using the *SegNet* [12] semantic segmentation on real image data from *StreetView* demonstrates the effectiveness of our algorithm's full pipeline on real data, resulting in better performance than the best reference algorithm. Despite the high performance, our system still receives a moderate amount of false localizations, which is due to similar sub-graphs at different locations, and we hope to mitigate this effect by including it into a full SLAM system in the future.

Furthermore, *3D* locations of the vertices are presently positioned at the blob centers of their first observation. Although, we split large objects at a maximum distance in several objects, the locations remain slightly different for different view points. We expect a more precise positioning technique to further disambiguate the associations between graphs, and increase localization accuracy.

V. Conclusions

In this paper we presented *X-View*, a multi-view global localization algorithm leveraging semantic graph descriptor matching. The approach was evaluated on one real-world and two simulated urban outdoor datasets with drastically different view-points. Our results show the potential of using graph representations of semantics for large-scale robotic global localization tasks. Alongside further advantages, such as compact representation and real-time-capability, the presented method is a step towards view-point invariant localization.

Our current research includes the investigation of more sophisticated graph construction, and matching methods, the integration of *X-View* with a full SLAM system to generate loop closures, and learning-based class selection for discriminative representations.

References

[1] M. Cummins and P. Newman, “Fab-map: Probabilistic localization and mapping in the space of appearance,” *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[2] D. Gálvez-López and J. D. Tardos, “Bags of binary words for fast place recognition in image sequences,” *IEEE Transactions on Robotics*, vol. 28, no. 5, pp. 1188–1197, 2012.

[3] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, “Visual place recognition: A survey,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.

[4] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: Cnn architecture for weakly supervised place recognition,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5297–5307.

[5] A. Gawel, T. Cieslewski, R. Dubé, M. Bosse, R. Siegwart, and J. Nieto, “Structure-based vision-laser matching,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 182–188.

[6] A. Gawel, R. Dubé, H. Surmann, J. Nieto, R. Siegwart, and C. Cadena, “3d registration of aerial and ground robots for disaster response: An evaluation of features, descriptors, and transformation estimation,” in *IEEE International Symposium on Safety, Security*, 2017.

[7] Z. Chen, A. Jacobson, N. Sunderhauf, B. Upcroft, L. Liu, C. Shen, I. Reid, and M. Milford, “Deep learning features at scale for visual place recognition,” 2017.

[8] E. Stumm, C. Mei, S. Lacroix, J. Nieto, M. Hutter, and R. Siegwart, “Robust visual place recognition with graph kernels,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4535–4544.

[9] Y. Su, F. Han, R. E. Harang, and X. Yan, “A fast kernel for attributed graphs,” in *SIAM International Conference on Data Mining*, 2016, pp. 486–494.

[10] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, “A review on deep learning techniques applied to semantic segmentation,” *arXiv preprint arXiv:1704.06857*, 2017.

[11] A. Valada, J. Vertens, A. Dhall, and W. Burgard, “Adapnet: Adaptive semantic segmentation in adverse environmental conditions,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4644–4651.

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[13] S. A. Cook, “The complexity of theorem-proving procedures,” in *ACM symposium on Theory of computing*. ACM, 1971, pp. 151–158.

[14] M. J. Milford and G. F. Wyeth, “Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1643–1649.

[15] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, “Point cloud descriptors for place recognition using sparse visual information,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 4830–4836.

[16] M. Bürki, I. Gilitschenski, E. Stumm, R. Siegwart, and J. Nieto, “Appearance-based landmark selection for efficient long-term visual localization,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4137–4143.

[17] N. Sunderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford, “Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free,” *Robotics: Science and Systems*, 2015.

[18] W. H. Huang and K. R. Beevers, “Topological map merging,” *The International Journal of Robotics Research*, vol. 24, no. 8, pp. 601–613, 2005.

[19] D. Marinakis and G. Dudek, “Pure topological mapping in mobile robotics,” *IEEE Transactions on Robotics*, vol. 26, no. 6, pp. 1051–1064, 2010.

[20] I. Kostavelis and A. Gasteratos, “Semantic mapping for mobile robotics tasks: A survey,” *Robotics and Autonomous Systems*, vol. 66, pp. 86–103, 2015.

[21] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, “Probabilistic data association for semantic slam,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.

[22] N. Atanasov, M. Zhu, K. Daniilidis, and G. J. Pappas, “Semantic localization via the matrix permanent.” in *Robotics: Science and Systems*, 2014.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” 2014, pp. 701–710.

[24] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection

of synthetic images for semantic segmentation of urban scenes," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

[25] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017.

[26] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.