2012 IEEE International Conference on Robotics and Automation (ICRA 2012).
May 14-18, 2012, St Paul, Minnesota, USA.

Real-Time Topometric Localization

Hernán Badino, Daniel Huber and Takeo Kanade
The Robotics Institute, Carnegie Mellon University
Pittsburgh, PA, USA

***Abstract*— Autonomous vehicles must be capable of localizing even in GPS denied situations. In this paper, we propose a real-time method to localize a vehicle along a route using visual imagery or range information. Our approach is an implementation of topometric localization, which combines the robustness of topological localization with the geometric accuracy of metric methods. We construct a map by navigating the route using a GPS-equipped vehicle and building a compact database of simple visual and 3D features. We then localize using a Bayesian filter to match sequences of visual or range measurements to the database. The algorithm is reliable across wide environmental changes, including lighting differences, seasonal variations, and occlusions, achieving an average localization accuracy of 1 m over an 8 km route. The method converges correctly even with wrong initial position estimates solving the kidnapped robot problem.**

I. Introduction

Localization is a central problem for autonomous vehicles that is often addressed using global positioning systems (GPS), inertial navigation systems (INS), or both. At times, the GPS signal may be unavailable, due to occlusions from buildings in cities or trees in suburban areas, or due to jamming from adversaries in military situations. Even when available, the GPS signal accuracy is limited, with errors of 3 to 6 m a common occurance, and INS devices, while able to improve accuracy and bridge short-term GPS dropouts, are expensive. These observations motivate the need for cost-effective, long-term localization in GPS-denied situations.

In this paper, we develop a long-term localization algorithm that builds on the visual topometric localization approach proposed in [4]. Most existing visual localization approaches fall into two categories: metric or topological. Metric localization produces exact measurements of the observer's pose or position on a map, typically using triangulation [14] or alignment [5]. Topological localization estimates the observer's location qualitatively from a finite set of possible positions. Such methods typically operate on a graph in which nodes represent possible locations and edges connect adjacent locations. Each approach offers complimentary advantages: metric localization can be very accurate, but may drift over time or fail for long sequences, while topological localization may be more reliable, but only provides rough position estimates.

Topometric localization combines topological and metric localization to achieve the reliability of topological localization with the geometric accuracy of metric localization. Topometric localization uses the same graph structure of topological localization, but at a fine-grained level. Associated with each node in the graph is a real, metric location. Unlike topological methods, the nodes are not semantically meaningful and are not necessarily distinguishable using the available sensors. Topometric localization involves two tasks: creating the map and localizing with respect to the map.

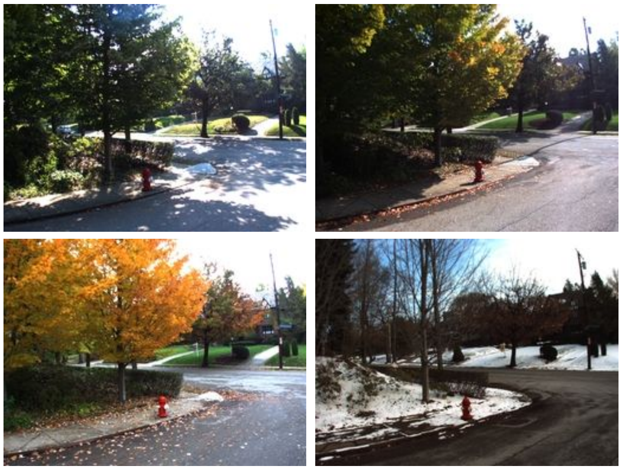
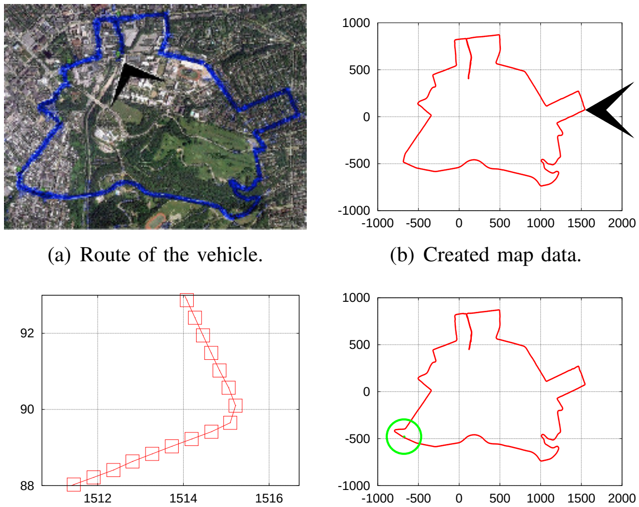


Fig. 1. Localization can be difficult in varying natural environments, such as this viewpoint seen during different seasons.

Map creation. A vehicle equipped with sensors and a GPS travels the route to be recognized one or more times. This may seem like a daunting task, but the Google Streetview project [8] shows that such mapping is feasible. In our algorithm, as the vehicle travels the route, a graph is created using the vehicle position at fixed distance intervals, and the nodes are annotated with the vehicle position. Visual or 3D features are extracted from the available sensor data and stored in a database along with a reference to the corresponding graph node. We currently focus on non-branching routes. Branching routes could be addressed by separating the map into non-branching segments, and a more general approach is the subject of ongoing research.

Localization. At runtime, as the vehicle drives over the mapped routes, a Bayes filter is used to estimate the probability distribution of the vehicle position along the route by matching features extracted from the sensor data to the map's feature database. Although we describe and evaluate the algorithm using an autonomous car, the method is equally applicable to indoor localization tasks.

Localizing based solely on visual appearance or range measurements presents a number of challenges. The algo-



(a) Route of the vehicle.

(b) Created map data.

(c) Corner marked in 2b

(d) Ground truth data of test run.

Fig. 2. Created map and ground truth data for evaluation (units in meters). The arrow in 2a shows the start and end position. Figure 2c amplifies the section marked in 2b. The circled area in 2d shows the location of the construction site (see text for details).

rithm must handle environmental changes over time (e.g., seasonal changes, building construction), lighting differences (e.g., sunny or cloudy days), and occlusions by other vehicles on the road or parked. Figures 1 and 10 show examples of all these cases. Furthermore, the vehicle may not follow the exact same route each time, driving, for example, in different lanes each time. The algorithm must also be robust to small changes in the route.

This paper contains four contributions with respect to our previously published work on topometric localization [4]. 1) We develop a new global visual feature that reduces the data storage requirements by a factor of 200 and increases the speed of the localization by more than 15 times, thereby enabling real-time localization (Section III-D). 2) We propose new 3D features based on simple statistics from a single line laser scanner that, when combined with the visual features, improve localization convergence (Section III-D). 3) We analyze the algorithm's performance for solving the kidnapped robot problem (Section V-B). 4) We evaluate our algorithm across data sets collected over a period of seasonal change (Section V-C).

II. Related Work

We will give here only a brief overview of the most related approaches.

Most localization methods rely on the extraction of local features from images and range data to build a visual database of the environment. Valgren and Lilienthal [17] evaluated the use of SIFT and SURF features for long-term topological localization. The authors used high-resolution panoramic images acquired over long periods of time to capture the natural seasonal changes in an outdoor environment. A comparison of results using variants of the SURF and SIFT features was performed. Ascani et al. [3] also evaluated the relative performance of SURF and SIFT features, finding that SIFT performs better in indoor environments. SIFT features were also used indoors using a topological approach by Andreasson et al. [2] and a metric approach by Se et al. [13]. Murillo et al. [12] proposed a two step approach. First, SIFT features were used to obtain the topological location of the observer. Second, a refinement of the location was obtained by computing the trifocal tensor between the best and second best database image matches. Silveira et al. [14] proposed a metric localization approach without explicit visual local feature detection. Although all these methods show promising localization results, the problem was approached as purely topological or metric, and the size of the maps varied from small (a few dozen images) to relatively small (less than 2500 images), whereas our experiments use databases on the order of $26,000$ images.

The fusion of topological and metric localization has also been approached before mainly in the simultaneous localization and mapping (SLAM) domain and using 3D sensors. Tomatis et al. [16], Kouzoubov and D. Austin [10], Bosse et al. [6], and Blanco et al. [5] used hybrid approaches to connect local metric submaps using high level topological maps. In all cases, the fusion aims at segmenting metric maps represented by topological nodes in order to organize and identify submap relations and loop closures.

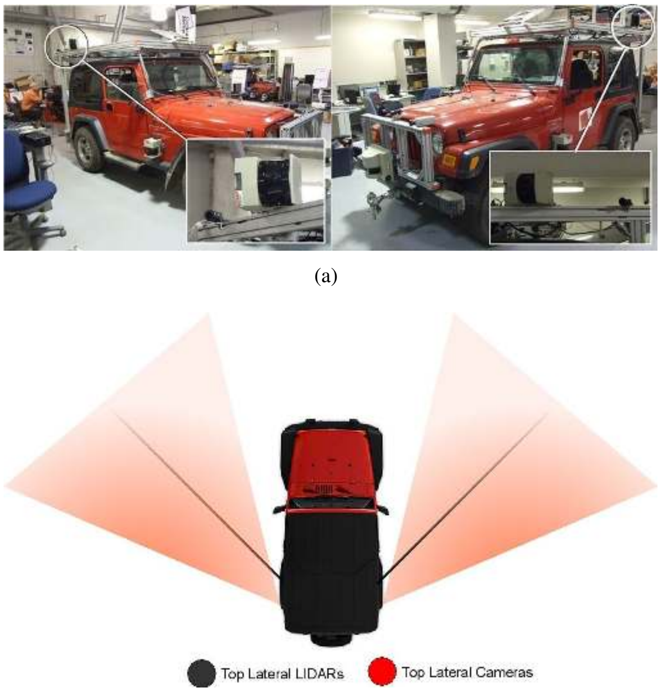
Our method differentiates from other approaches principally in two ways. 1) Instead of obtaining large number of complex features extracted from image and range data, our method relies on simple statistics of the observations, allowing us to minimize computation and storage requirements. As we will see in the experimental results, reliable localization can be achieved by the proper integration of the measurements using Bayesian estimation. 2) Instead of creating 3D maps or topological structures based on clustering, our method integrates metric data directly into the topological nodes by creating a database of the route, as adresssed in the next section.

III. Topometric Localization

In this section, we describe the two components of the topometric localization algorithm: map creation and localization. Additional details may be found in [4].

A. Map Creation

The route map is a directed graph in which the nodes represent possible vehicle locations and the edges represent transitions between locations (Figure 2b). As the vehicle drives the route, a new node storing the vehicle's current location is added each time the vehicle moves a constant Euclidean distance ρ from the location of the previous vertex (Figure 2c), and a vertex connecting the two nodes is also added. At the same time, features are extracted from the sensors and stored in a database. Each extracted feature is stored with a reference to the current map node. In this work, we evaluate features derived from visual imagery and range information. For visual sensing, we use two cameras aimed 45° to the left and right of forward (Figure 3a). Two vertically scanning single line lidars provide range



(a)

(b)

Fig. 3. Evaluation vehicle

information in "push-broom" format as the vehicle moves forward.

A number of feature descriptors can be used for localization [7]. We have experimented with several types of features extracted from visual imagery and range data. In [4], we used upright SURF (U-SURF) descriptors. Here, we introduce a whole image descriptor as well as two range-based descriptors:

- *Whole image SURF (WI-SURF) descriptor*. A vector $\boldsymbol{d} \in \mathbb{R}^{64}$ containing gradient information of the entire image, computed as described in [1] (Figure 4).
- *Range mean* and *range standard deviation*. The average $\bar{r}$ and standard deviation σ of all range measurements acquired during the cycle time Δt, where Δt is the time between consecutive image acquisitions.

The measurement database consists of the set $\mathcal{D} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_M\}$ with components $\boldsymbol{r}_i = \{\boldsymbol{d}_i, \bar{r}_i, \sigma_i, l_i\}$, where l_i is the ground truth location of the vehicle in the map.

The WI-SURF descriptor provides a significant benefit over the previously reported U-SURF descriptors in terms of database size and runtime efficiency. Using U-SURF descriptors, a typical feature database for our test route contains over two million descriptors, since each image may result in several detected features. The WI-SURF descriptor only requires one feature per image, reducing the database from approximately 1 GB to 5 MB (a factor of 200 reduction). The WI-SURF descriptor also significantly reduces computational requirements. The standard SURF/SIFT matching algorithm requires $n!/(2(n-2)!)$ vector comparisons [11] ($19,900$ comparisons per image for $n = 200$), whereas WI-SURF matching requires only one descriptor comparison per image. This optimization gives us a speed up factor of 15 with respect to the U-SURF approach. The experiments in Section V-A demonstrate that the localization using the WI-SURF descriptor has the same level of accuracy as achieved using the U-SURF descriptors. Our current implementation of the localization algorithm runs at 15 Hz on a standard laptop.

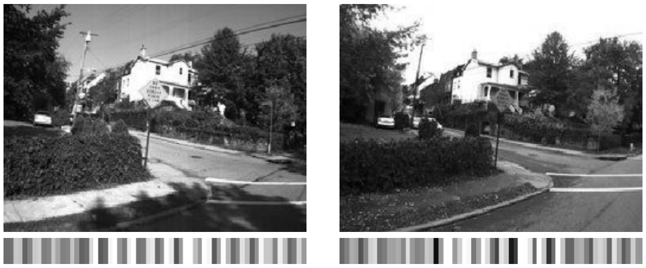


Fig. 4. Example of corresponding images (top) and grayscale representation of their descriptors (bottom)

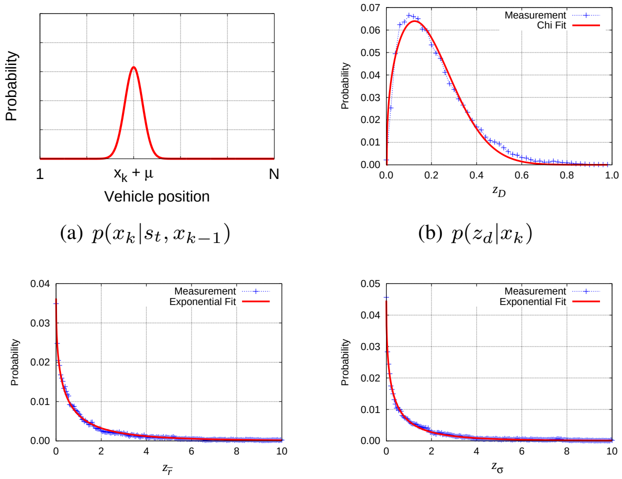
B. Bayesian Localization

The state X_t defines the position of the vehicle in the map at time t (i.e., the graph vertex where the vehicle is located). The vertices of the graph define the possible values that X_t might assume, i.e., $x_k, k = 1, 2, 3, ..., N$. The probability of the vehicle located at a particular position x_k in the graph at time t is specified as $p_{k,t} = p(X_t = x_k)$. The discrete probability distribution is expressed as the set $p(x_k) = \{p_{k,t}\}$. The Bayes filter [15] keeps track of the probabilities as the vehicle moves and new measurements are acquired.

Two actions modify the pdf: prediction and update. The prediction is the incorporation of the known motion of the vehicle into the previous pdf. The update step incorporates new measurements into the current predicted pdf. The discrete Bayes filter performs these two steps recursively:

```
**Input**: $\{p_{k,t-1}\}, s_t, z_t$
**Output**: $\{p_{k,t}\}$
# Predict
$\bar{p}_{k,t} = \sum_{i=1}^{N} p(X_t = x_k | s_t, X_{t-1} = x_i)\; p_{i,t-1}$;
# Update
$p_{k,t} = \eta\; p(z_t | X_t = x_k)\; \bar{p}_{k,t}$
```

The *state transition probability* $p(X_t = x_k | s_t, X_{t-1} = x_i)$ in the prediction step specifies how the state evolves over time as the vehicle moves with velocity s_t in the environment. The *measurement probability* $p(z_t | X_t = x_k)$ in the update step specifies how measurement probabilities are generated from the system state x_k. The scalar η ensures that the resulting pdf integrates to one. The discrete Bayes filter requires the definition of the state transition probability and the measurement probability as well as the initial probability density function $p_0 = \{p_{k,0}\}$. The following sections derive these functions for our localization problem.



(a) $p(x_k|s_t, x_{k-1})$

(b) $p(z_d|x_k)$

(c) $p(z_{\bar{r}}|x_k)$

(d) $p(z_\sigma|x_k)$

Fig. 5. State transition and measurement probability density functions

C. State Transition Probability

We will assume that the velocity of the vehicle s_t at time t is a zero-mean noisy measurement with variance σ_s^2. The translation of the vehicle in the map between times $t-1$ and time t is $\mu = s_t \Delta t / \rho$ with a variance of $\sigma^2 = (\Delta t\, \sigma_s / \rho)^2$, where Δt is the cycle time, and ρ is the distance between consecutive nodes in the map ($\rho = 1$ m in the experimental results of Section V). We model the state transition probability with a Gaussian pdf (Figure 5a):

$$p(x_{k,t}|s_t, x_{i,t-1}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_{i,t-1} - \mu)^2}{2\sigma^2}\right) \quad (1)$$

Equation 1 has the effect of translating and smoothing the vehicle location pdf $p(x_{k,t-1})$.

D. Measurement Probability

The measurement probabilities are functions of the similarity between the extracted features and those in the database. To convert the similarity measures into probabilities, we compute empirical probability distributions from sample data and fit parametric models to the empirical pdfs. This approach is in contrast to [4], where the pdfs were custom-designed based on theoretical assumptions.

We define the dissimilarity between a measured WI-SURF descriptor $\boldsymbol{d}$ and a database descriptor $\boldsymbol{d}_i$ using the norm of the descriptor difference, i.e.,

$$z_d = |\boldsymbol{d} - \boldsymbol{d}_i| \quad (2)$$

Since the image descriptor vectors are normalized, z_d lies in the interval $[0, 1]$. Similarly, we define the distance between current range features $\bar{r}$ and σ and database range features $\bar{r}_i$ and σ_i as the square root difference between observed quantities, i.e.,

$$z_{\bar{r}} = |\bar{r} - \bar{r}_i| \quad (3)$$

and

$$z_\sigma = |\sigma - \sigma_i|. \quad (4)$$

The pdf of the measurements $p(z|x_k)$, which is needed by the Bayes filter, can be estimated from sample ground truth data. Figure 5b-d show the empirical distribution of the three dissimilarity measures for the data sets described in Section IV-D. We observe that the distributions closely match common parametric distributions. The WI-SURF dissimilarity (z_d) follows a Chi-squared distribution (Figure 5b). The reduced probability for very small difference values is primarily due to positioning offsets, lighting variation, seasonal changes, and other image differences that prevent perfect matching even at the correct location. The range mean ($z_{\bar{r}}$) and range standard deviation (z_σ) are well described by exponential distributions (Figure 5c and d respectively).

The learned models do not account for coincidental good matches that can occur between the measurement and database entries for other (incorrect) vehicle locations. Such matches occur, for example, due to repeated structures in the environment – a long, uniform building facade or the same model car parked in two locations along the route. We model these random occurrences with a uniform pdf over all possible vehicle locations. The final measurement pdf is a linear combination between the learned measurement pdf and the uniform distribution, i.e.,

$$p(z_d|x_k) = \eta_d \begin{cases} \alpha_d + \chi(z_d, k) & \text{if } x_k = l_i \\ \alpha_d & \text{otherwise} \end{cases} \quad (5)$$

$$p(z_{\bar{r}}|x_k) = \eta_{\bar{r}} \begin{cases} \alpha_{\bar{r}} + f(z_{\bar{r}}, \lambda_{\bar{r}}) & \text{if } x_k = l_i \\ \alpha_{\bar{r}} & \text{otherwise} \end{cases} \quad (6)$$

$$p(z_\sigma|x_k) = \eta_\sigma \begin{cases} \alpha_\sigma + f(z_\sigma, \lambda_\sigma) & \text{if } x_k = l_i \\ \alpha_\sigma & \text{otherwise} \end{cases} \quad (7)$$

In these equations, l_i is the location of the database feature used to compute the measurement; $\chi(z_d, k)$ is the learned Chi-squared distribution with k degrees of freedom; $f(z, \lambda)$ is the exponential pdf with corresponding learned rate parameter λ; α_d, $\alpha_{\bar{r}}$, and α_σ are factors weighting the strength of the uniform distributions; and η_d, $\eta_{\bar{r}}$, and η_σ are normalization factors that ensure that the pdfs integrate to one.

Assuming independence of the measurements, the combined measurement pdf for a time step t is simply the multiplication of the individual distributions:

$$p(z_t^i|x_k) = \eta_1 p(z_d^i|x_k)\, p(z_{\bar{r}}^i|x_k)\, p(z_\sigma^i|x_k) \quad (8)$$

where $z_t^i = (z_d^i, z_{\bar{r}}^i, z_\sigma^i)^T$ is the i^{th} measurement, and η_1 is a normalization factor. For M measurements in the database, the final probability function is obtained by the product of the individual measurement pdfs, i.e.,

$$p(z_t|x_k) = \eta_2 \prod_{i=1}^{M} p(z_t^i|x_k). \quad (9)$$

with normalization factor η_2.

E. MAP Estimation

The estimated location of the vehicle at every time step will be that with the largest probability within the set of possible locations, i.e., the maximum a posteriori (MAP) estimate. From the update equation of the Bayes filter algorithm, the MAP estimate is

$$X_t = \arg\max_k (p_{k,t}). \tag{10}$$

IV. Experimental Setup

We conducted our experiments using a test vehicle with mounted video cameras and LIDARs operating on an 8 km test route under varying environmental conditions.

A. Vehicle

Figure 3 shows our evaluation vehicle sensor suite. Two cameras and two lidars were mounted on the roof of the vehicle, oriented approximately 45° to the left and right of directly forward as shown in Figure 3b. The diagonal orientation was chosen considering the probability of observing moving objects, and the expected amount of change in the appearance of the environment while driving. The cameras were configured to acquire 256×192 pixels images at 15 Hz. The lidars acquire range measurements at 75 Hz with field of view of 180 degrees and angular resolution of one degree. The vehicle is also equipped with a differential GPS and an advanced inertial motion unit system for map generation and acquisition of ground truth data.

B. Evaluation Route

For the purpose of analysis and evaluation, we selected a complex, 8 km route that contains a variety of environments, ranging from urban to residential to parklike settings (Figure 2a). The route includes man-made and natural structures: buildings, traffic signs, foliage, open spaces, and multiple slopes, as well as moving objects such as vehicles, pedestrians, cyclists, and motorcyclists. The trajectory makes a loop, meaning that the vehicle must face all orientations during the trajectory. This is important in order to test the robustness of the method to illumination artifacts, such as specularities made by direct sunlight exposure.

C. Generation of Ground Truth Data

The route map and ground truth information were mainly obtained by the GPS system, which provides a zero-mean estimation of the vehicle position (Figure 6a). However, GPS data is noisy, it is obtained at a low rate (1 Hz), and it might be incorrect or missing when the signal is blocked by buildings or trees. On the other hand, the inertial motion system is always available, is obtained at a high framerate (80 Hz), and is very accurate for short distances. In contrast to GPS, the position of the vehicle as computed by the inertial sensor drifts over long periods of time (Figure 6b). In order to compensate for each sensor deficiencies, we combine both positioning estimates obtaining dense and accurate vehicle trajectory. The fusion is performed by locally aligning the GPS and odometry trajectories. For this purpose, GPS positions are first interpolated to match the density of the odometry data (Figure 6c). Then, for each interpolated GPS position (green dot in Figure 6c), a local neighborhood of the GPS path (typically ±400 m) is aligned with the odometry path (Figure 6e). A robust version of the absolute orientation problem [9] is used for finding the transformation between the data sets. The data association is obtained thanks to accurate data timestamping (Figure 6d). Finally, the new ground truth path position is obtained from the corresponding aligned odometry point (large yellow dot in Figure 6f).

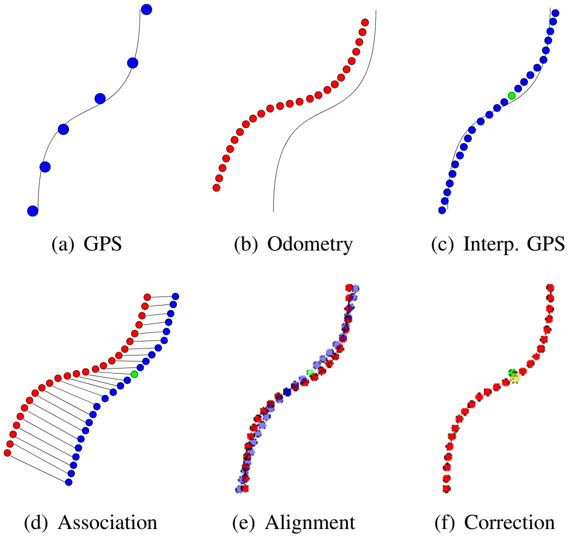

(a) GPS (b) Odometry (c) Interp. GPS
(d) Association (e) Alignment (f) Correction

Fig. 6. Example of GPS/Odometry fusion. The continuous line shows the ground truth trajectory.

D. Data Sets

We acquired five data sets on the test route over a period of 4 months between September and December, 2010 (Table I). The October 28 data set was used to generate the measurement database (first column of Figure 10). The effects of the fall season are visible orange-colored vegetation and leaves on the ground. Two databases $\mathcal{D}_L$ and $\mathcal{D}_R$ for the left and right sensors sets (Figure 3a) were generated offline. Each entry of the databases contains a WI-SURF descriptor $\boldsymbol{d}_i$, the range mean and standard deviation $\bar{r}$ and σ, and the vertex number l_i of the map corresponding to the observed features. An independent Bayes filter is initialized for the left and right sensors. The final pdf is obtained by the fusion of left and right probabilities, i.e.:

$$p(x_t) = \eta_3\, p_L(x_t) p_R(x_t). \tag{11}$$

where $p_L(x_t)$ and $p_R(x_t)$ are the left and right pdfs respectively, and η_3 is a normalization factor.

V. Experimental Results

In this section, we conduct three experiments to evaluate the performance of our localization algorithm. In the following experiments, please note that GPS and odometry were only used for the generation of the map and ground truth data.

A. WI-SURF Descriptor versus U-SURF Descriptor

In this section, we evaluate the performance of the WI-SURF descriptor in relation to the previous U-SURF descriptor approach [4]. Figure 7 shows the localization results

Date	Images	Particularities
September 1, 2010, 9:41 AM	$11,140\times 2$	Sunny, abundant green vegetation, strong large shadows.
October 19, 2010, 11:58 AM	$11,857\times 2$	Sunny, abundant colored vegetation, shadows.
October 28, 2010, 11:23 AM	$13,357\times 2$	Cloudy, reduced colored vegetation, fallen leaves.
November 22, 2010, 11:52 AM	$13,319\times 2$	Cloudy. Reduced vegetation, dim lighting.
December 21, 2010, 14:17 AM	$11,250\times 2$	Cloudy. Reduced vegetation, snow on ground.

TABLE I: Data sets used in the experiments

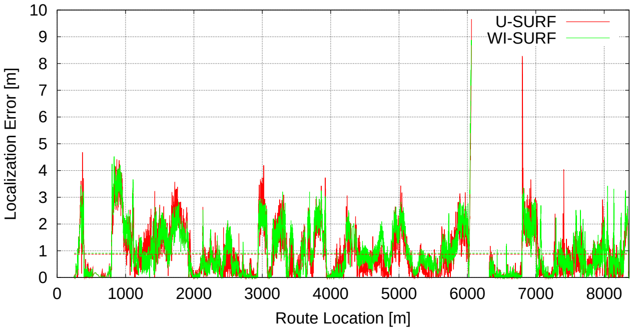


Fig. 7. Localization results using U-SURF and WI-SURF features

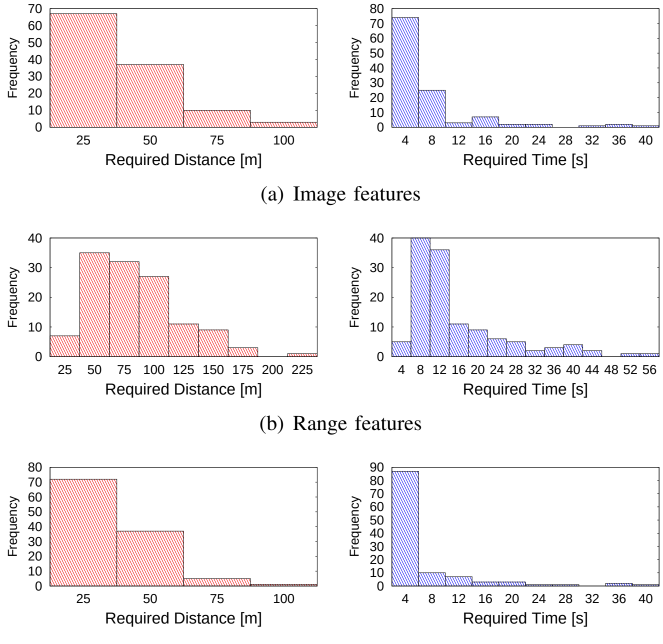
using the October 19 dataset. The initial probability density function $p_0 = \{p_k, 0\}$ was initialized with a point mass distribution on the ground truth position of the vehicle at the start of the route. Given the long time required by the U-SURF descriptor approach, only measurements at a maximum distance of ±80 nodes (i.e., ±80 m) from the current estimated position of the vehicle were considered in this experiment. The location of the vehicle at each time step is obtained from the MAP of the estimated pdf. Range features were not used in this experiment.

The results between both approaches were practically equivalent as the following table shows.

	Avg Error	Std Dev	Max Error	Div
U-SURF Descriptor	$0.87m$	$0.97m$	$8.28m$	0
WI-SURF Descriptor	$0.91m$	$0.84m$	$4.53m$	0

Even though the average error slightly increases, it is better bounded and its standard deviation decreases. No divergences (last column of the table) occurred throughout the sequence, i.e., the algorithm stayed converged all the time. The missing gap between segments 6000 and 6200 in Figure 7 was produced because the vehicle drove slightly different routes. A short street of our predefined route was closed because of a construction site in the south-west part of the route (see Figs 2d and 10e). We have increased the tracking area in this part of the sequence to ±480 m in order to allow the Bayesian method to find the actual location once the vehicle rejoins its original route.

Our new WI-SURF provides a powerful alternative for large scale vehicle localization, requiring small storage and computation requirements. Our current implementation performs *global* tracking at 5 Hz, in contrast to the 5 Hz *local* tracking (±80 m window) that we achieved using the U-SURF descriptor approach.



(a) Image features

(b) Range features

(c) Image and range features

Fig. 8. Distance and time of travel required by the algorithm to converge to the ground truth location solving the kidnapped robot problem

B. Kidnapped Robot Analysis

Global localization, in contrast to position tracking, implies localizing the vehicle considering the full state space of possible locations. Global localization is a desirable capability for an autonomous robot system, because it is important to ensure that the robot can recover even if incremental position tracking fails.

We have performed an experiment to evaluate the capability of our algorithm to solve the kidnapped robot problem. In this experiment, we repeatedly "teleport" the vehicle to a random route position. After each teleportation, we just let the vehicle continue its standard global localization task, estimating the full pdf of the vehicle location along the route. We identify that the vehicle has been re-localized when the MAP estimate corresponding to its ground truth location is at least ten times larger than the second maximum peak in the pdf. We ensure, in this way, not only the convergence of the estimation to the new real location, but also that the vehicle starts with a very strong confidence on a wrong location as soon as it is teleported again. This means that after the teleportation, the vehicle must lose its gained position belief and build up new one to re-localize itself.

Figure 8 shows the time of travel and distance required by the vehicle to find its new ground truth location using

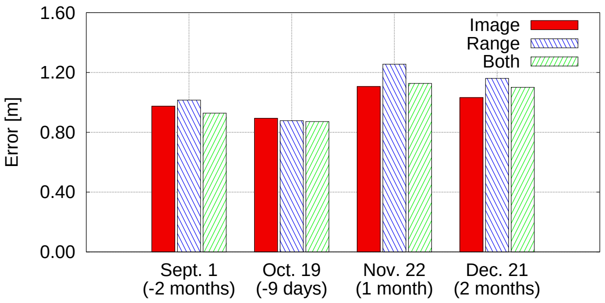


Fig. 9. Average localization errors

the data set acquired on October 19. We have evaluated the relocalization performance by using the image and range features independently and in combination. Between 115 and 125 teleportation were performed in each case. Image features provide more reliable location evidence than range features, allowing the vehicle to re-localize faster in shorter distances. The best relocalization strategy is given by using both image and range features. Observe that most of the time not more than 50 meters and 4 seconds of travel are required by the algorithm to achieve the relocalization. It is important to remark that the algorithm was successful every time, i.e. 100% convergence success.

C. Long Term Localization

In this experiment, we evaluate the relative performance of the extracted features on all data sets. The results are shown in Table II, and Figures 9 and 10. The smallest localization error is found for the closest dataset (Oct. 19), but the error does not seems to increase as a function of elapsed time. Range measurements provide, on average, slightly worse localization performance than image features, and the fusion of image and range features does not necessarily improve the localization accuracy (it does, however, improve the convergence rate, as demonstrated in the previous experiment). In general, the localization was reliably computed along all data sets. The largest average localization error is 1.26 m with absolute maximum localization error of 7.68 m.

The last column of Table II shows the number of divergences in each case. A divergence is detected if the localization error is more than 10 m. Divergences usually occur due to long-lasting occlusions, such as when vans or large trucks drive parallel to the vehicle, occluding visualization almost completely (first image on Figure 10d). In all cases, the algorithm had no problems to converge rapidly again to the right solution once the situation normalized. We have also tested the performance of the algorithm by tracking the vehicle (instead of performing global localization). The tracking interval was set to 480 m around the current position estimate. The localization errors obtained were the same as shown in Table II (and therefore omitted for brevity), but no divergences occurred.

VI. Conclusions

We have demonstrated that the real-time localization of vehicles can be achieved using a combination of simple statistics of the measured environment and the appropriate Bayesian estimation. A simple pair of low resolution cameras is more than enough to perform accurate localization even over long periods of time on varying environmental conditions. When possible and available, the use of range information, can further help the localization task by providing not only redundancy, but also improved localization convergence.

Data set	Meas	Avg Error	Std Dev	Max Error	Div
Sept. 1	Image	0.97 m	1.08 m	4.34 m	1
	Range	1.01 m	1.11 m	6.23 m	0
	Both	0.92 m	0.99 m	6.23 m	0
Oct. 19	Image	0.89 m	0.84 m	4.34 m	1
	Range	0.88 m	0.95 m	5.07 m	1
	Both	0.87 m	0.82 m	4.39 m	1
Nov. 22	Image	1.11 m	1.39 m	6.28 m	0
	Range	1.26 m	1.41 m	7.68 m	0
	Both	1.13 m	1.38 m	6.28 m	0
Dec. 21	Image	1.03 m	1.14 m	5.15 m	1
	Range	1.16 m	1.22 m	6.66 m	3
	Both	1.10 m	1.20 m	5.76 m	3

TABLE II: Localization results

VII. Acknowledgments

This work was supported by the Agency for Defense Development, Jochiwongil 462, Yuseong, Daijeon, Korea.

References

[1] M. Agrawal, K. Konolige, and M. R. Blas. Censure: Center surround extremas for realtime feature detection and matching. In *eccv*, volume 5305, pages 102–115, 2008.

[2] H. Andreasson, A. Treptow, and T. Duckett. Localization for mobile robots using panoramic vision, local features and particle filter. In *ICRA*, 2005.

[3] A. Ascani, E. Frontoni, and A. Mancini. Feature group matching for appearance-based localization. In *IROS*, 2008.

[4] H. Badino, D. Huber, and T. Kanade. Visual topometric localization. In *Intelligent Vehicles Symposium*, 2011.

[5] J. Blanco, J. Fernández-Madrigal, and J. González. Toward a unified bayesian approach to hybrid metric-topological slam. *IEEE Transactions on Robotics*, 24(2):259–270, 2008.

[6] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An atlas framework for scalable mapping. In *icra*, volume 2, pages 1899–1906, 2003.

[7] A. Cumani and A. Guiducci. Comparison of feature detectors for rover navigation. *Mathematical Modelling and Applied Computing*, 1(1), 2010.

[8] http://maps.google.com/help/maps/streetview/

[9] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. In *Journal of the Optical Society of America A*, volume 4(4), pages 629–642, April 1987.

[10] H. Kouzoubov and D. Austin. Hybrid topological/metric approach to slam. In *ICRA*, volume 1, pages 872–877, 2004.

[11] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Robotics Research*, 2:91–110, 2004.

[12] A. C. Murillo, J. J. Guerrero, and C. Sagüés. Surf features for efficient robot localization with omnidirectional images. In *ICRA*, 2007.

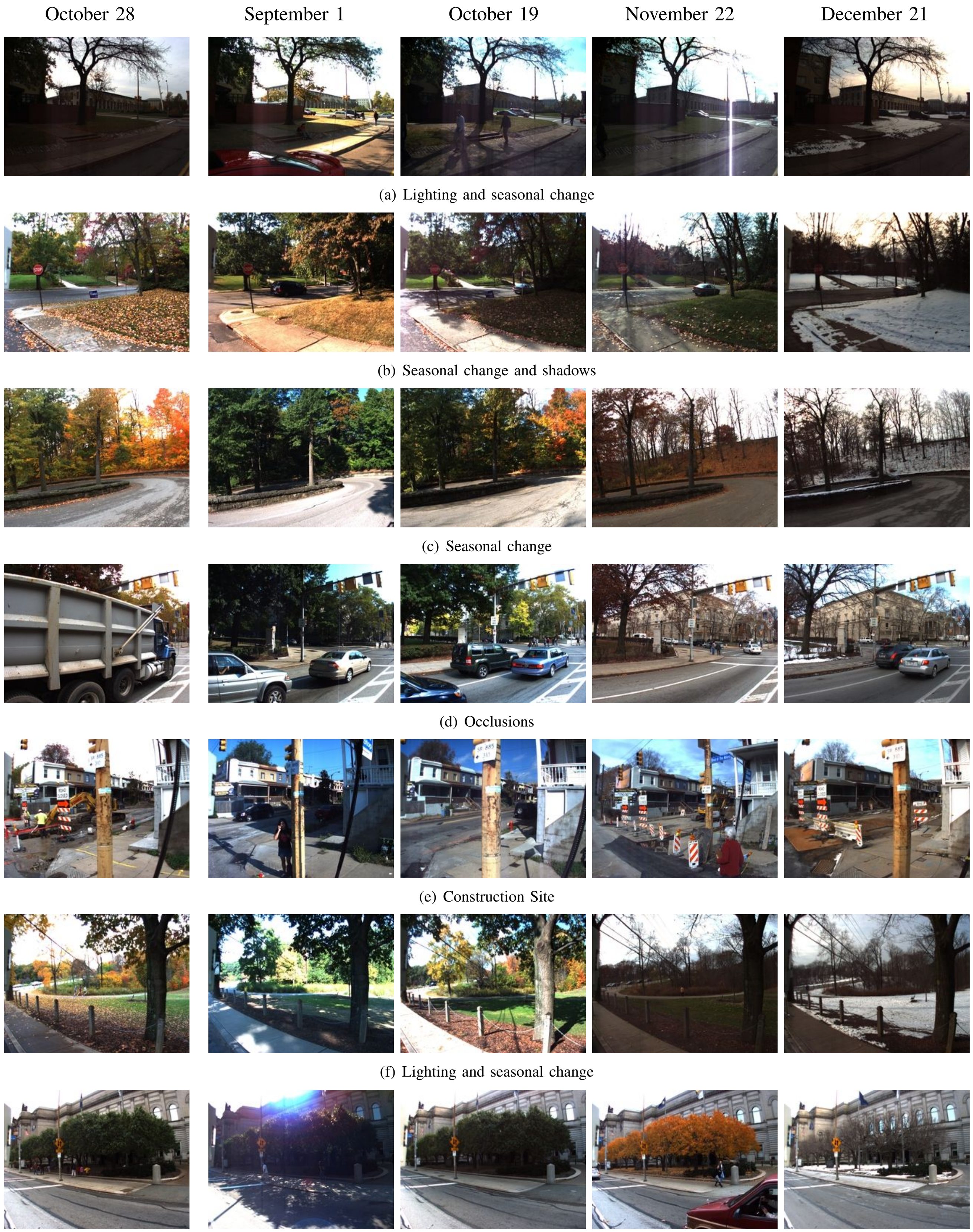
[13] S. Se, D. G. Lowe, and J. J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 2005.

[14] G. Silveira, E. Malis, and P. Rives. An efficient direct approach to visual slam. *IEEE Transactions on Robotics*, 2008.

[15] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005.

[16] N. Tomatis, I. Nourbakhsh, and R. Siegwart. Hybrid simultaneous localization and map building: a natural integration of topological and metric. *Robotics and Autonomous Systems*, 3(14), 2003.

[17] C. Valgren and A. J. Lilienthal. Sift, surf & seasons: Appearance-based long-term localization in outdoor environments. *Submitted to Elsevier*, 2010.



(a) Lighting and seasonal change

(b) Seasonal change and shadows

(c) Seasonal change

(d) Occlusions

(e) Construction Site

(f) Lighting and seasonal change

(g) Lighting, shadows, and seasonal change

Fig. 10. Example of localization at different route positions. The appearance changes because of lighting, occlusions, seasonal changes, or a combination of these factors.