# Robust Scene Stitching in Large Scale Mobile Mapping

Filip Schouwenaars[1]
filip.schouwenaars@student.kuleuven.be
Radu Timofte[12]
radu.timofte@esat.kuleuven.be
Luc Van Gool[12]
luc.vangool@esat.kuleuven.be

[1] VISICS, ESAT-PSI / iMinds
KU Leuven
Leuven, Belgium

[2] Computer Vision Lab, D-ITET
ETH Zürich
Zürich, Switzerland

**Abstract**

We provide a solution for the loop closure problem in an image-based mobile mapping context. A van drives through a city while taking images in multiple directions. Local feature matching in two stages detects when a particular site is revisited, in order to enforce correspondences between such images, that may have been taken with large time lapses in between. Our system relies on GPS but does not use odometric information. We extend the original image-to-image matching approach to a pose-to-pose matching approach, combining several images and achieving robust scene matching results. Parameter optimization is followed by extensive experiments. Our pipeline, which facilitates parallel execution, reaches matching rates higher than those reported for typical state-of-the-art algorithms. We also demonstrate robustness to odometric inconsistencies resulting from poor prior model build-up.

## 1 Introduction

Image-based mobile mapping is the process of collecting geospatial information from a mobile vehicle and combining this data to build a model of the surroundings. The results can be used in a wide variety of applications, from city modeling and road mapping to emergency response planning. We make use of a van that is equipped with several cameras, that are organized in stereo pairs (one pair looking forward, one backward, and two from the side).

This paper deals with the recurring issue of drift affecting triangulated points over time. A van equipped with stereo cameras collects recordings in an urban environment, simultaneously monitoring GPS information. Using Structure-from-Motion (SfM) techniques [8], the position of the van and the 3D coordinates of the surroundings are retrieved. The determination of the translation and orientation of the van's position is recursive: a slight drift can gradually build up to flawed localizations. One can rely on the GPS information to perform adjustments, but its accuracy and availability are not always adequate to yield a model with high precision. Yet, visual loop-closing – recognizing that a location is revisited – may help mitigate the issue. Indeed, some sites of the scanned area are bound to be visited multiple times, *e.g.* dead-end streets, crossroads or access roads. In our current system, model fragments that originate from different recording moments but actually correspond to the same

physical environment are not combined nor treated differently. This paper adds such loop closure.

Several things have to be kept in mind for loop closure. Since it concerns time-distant recordings, environmental conditions can have changed substantially: lighting conditions, weather, traffic (severe occlusions) and other temporary elements in the scene may all have changed. On top of this, also the viewing angles most probably differ quite a bit. Avoiding false positives is a must.

**Outline of the proposed technique.** Our approach has two main components. First, sites revisited over time have to be detected. This is simplified by clustering the GPS information while taking its inaccuracy into account (see 3.1). Within those clusters, van location pairs are selected that are expected to have produced overlapping views, i.e. locations that are close to each other. We use a Naive Bayes (NB) matching framework for this purpose (see 3.1). Since this is a computationally intensive operation, one focus is on speeding it up. The second subproblem is that of finding re-occurrences of the same physical points in the images coming from the location pairs. We start by *single pose matching* (see 3.2.1) and match points using the SURF [1] detector and descriptor among views taken from the same van position. Correspondences between stereo views are backprojected to 3D points, which results in a point cloud for every van pose. In a subsequent *cross-pose image matching*, the images from different van locations are matched, again using SURF, and matches are accepted if they correspond to points that were triangulated earlier (see 3.2.2). This step introduces putative links between the point clouds for the two locations. PROSAC [2], a prioritized RANSAC algorithm, is applied to robustly and efficiently calculate the transformation between the two point clouds. A set of correspondences results that links time distant recordings. The success of the cloud matching determines whether a loop closure has been found.

Hence, rather than matching single images, we match 3D point clouds. This said, in order to get there, we still have to solve features under wide baselines. Each 3D point is described by features from the originating images.

**Structure of the paper.** Section 2 revisits techniques for mobile mapping and feature detection and description. Section 3 describes the application in more detail, where the two aforementioned subproblems are elaborated on. We also focus on measures to reduce the computational load. Section 4 describes experimental results and the effects of the acceleration measures are investigated. Some perturbed mock-up datasets are tested to challenge the system. After an illustration of the actual embedding of our approach into the current mobile mapping system, the paper is concluded in Section 5.

# 2 Related Work

## 2.1 Mobile Mapping & Scene Stitching

A general description of the problem of mobile mapping, solution strategies and applications is given in [17]. Loop closure can be crucial if one want to produce highly accurate models. The FAB-MAP algorithm, a s-o-a topological mapping method [3] uses a bag of words approach to model locations in an appearance-based manner. By assigning a probability of an observation having come from a previously visited place, a fast yet robust system is developed that enables real-time mobile mapping. However, only 40% of re-occurrences

are detected, and these detections highly depend on the amount of occlusion and the driving direction. The same limitations apply to [21], that uses a holistic descriptor and an efficient matching scheme for recognition. Even stronger assumptions are made here, since the holistic BRIEF descriptor is sensitive to scaling and translation. FAB-MAP has been extended using graph theory [16] and faster implementations have been designed [4, 6].

CAT-SLAM [12] and its graphical extension CAT-GRAPH [13] augment this sequential appearance-based place recognition with local metric pose filtering to improve the frequency and reliability of appearance-based loop closure. The method shows similarities with FAB-MAP, but uses odometric information from previous results in order to increase the number of correct loop closures.

Both FAB-MAP and CAT-SLAM deliberately avoid to build a 3D map and settle with a binary decision, *i.e.* whether or not the location was visited already. In the envisioned application however, it is desirable to have a 3D reconstruction to better handle correspondences over longer time lapses. An approach closer to this goal, by directly attempting to match local features among images, is described in [18, 19, 20]. Typically, a post-processing step that prunes false positives is performed. The epipolar constraint is used in [18, 19]. This constraint does not completely guard against false positives however, since a correspondence in one image is only bound to lie on a line in the other image. [20] resorts to another, rather intuitive spatial consistency measure to check the consistency of matches.

## 2.2 Feature Detection & Description

In several stages of the algorithm, image matching is performed. Searching for discrete image point correspondences in a local manner mainly includes three steps [1]: interest point detection, description and matching. A detector should have high repeatability. The descriptor ought to be as distinctive as possible and robust to noise and geometric and photometric deformations. These vectors are then in a last step matched and putative correspondences are retained. Matching can follow several metrics and efficient implementations exist [15].

Following [14], Hessian-based detectors [10] are more stable than their Harris-based [7] counterparts. At the descriptor side, the Scale-Invariant Feature Transform (SIFT) [11] is a very popular descriptor due to its high degree of distinctiveness and computational efficiency. The Speeded-Up Robust Features (SURF) [1] descriptor is a further speed-up of the SIFT descriptor without compromising performance, with box-filters replacing the Difference-of-Gaussians (DoG).

# 3 Scene stitching

Globally, consider a set of evenly spaced locations - also called a pose set hereafter - $\{p_i\}_{i=1}^{n_p}$ with $n_p$ the number of locations or 'poses'. The set that comprises the pose set indices is denoted $P$. Each pose $p_{i\in P}$ is related to a rotation and translation with respect to the global coordinate system. An example of the poses' translations that are derived from a single recording is shown in Fig. 1, where each pose has a different color.

The van is equipped with eight cameras $\{c_j\}_{j=0}^{7}$, *i.e.* the four aforementioned stereo pairs $\{c_0,c_1\}$, $\{c_2,c_3\}$, $\{c_4,c_5\}$ and $\{c_6,c_7\}$. The cameras are calibrated with respect to each other (internally and externally). A single recording moment at a certain pose thus relates to eight images $I_{i,j}$, where $i \in P$ denotes the pose index and $j$ is the camera index.

## 3.1 Candidate Pose Pair Detection

**Region of interest extraction.** Subsets of the pose set, so called *routes* $R \subset P$, that occur around the same van locations but from van passages at sufficiently different times, are collected from the pose set. A route corresponds to one such different van passage. The search radius around which the system scans, depends on the maximal drift in the model and was set to 15m. In one detected cluster $C_q = \{R_k\}_{k=1}^{n_r}$ for the current query pose $q \in P$, all possible $n_r - 1$ route pairs with the route that is found first are made. No need to say that $n_r$ is typically low, like 2, sometimes 3 or more.

**Pose pair selection.** From a route pair $(R_k, R_l)$ one pose has to be selected from every route, resulting in a *cross-route pose pair*: CRPP $= \left(p_{\tilde{r}_k}, p_{\tilde{r}_l}\right)$, where $\tilde{r}_k \in R_k, \tilde{r}_l \in R_l$ are well-chosen pose indices. For a quick pruning, one could resort to the pose information to only take into consideration poses that are very close. Since we want to rely minimally on the actual extracted pose information however, all poses from one route should be checked with all poses from the other route in all possible orientation settings, in order not to miss out on any possible match; an enormous task.

In order to find a good candidate pose pair quickly, a Naive Bayes framework was developed. Exhaustively, for all images $I_{r_k,j}$ and $I_{r_l,j}$ linked to the poses, a severely downscaled image ($154 \times 203$ pixels) is described using $N_{\text{NB}}$ 64-dimensional Upright SURF descriptors [1]. For every possible pose combination, matching is performed for every possible image combination. If one thus has $|R_k|$ poses in the first route[1], $|R_l|$ poses in the second route, and 4 cameras per pose[2], this means $16 \times |R_k| \times |R_l|$ pairs must be matched, before a decision can be made on which CRPP is the most promising to continue with. The pose pair and specific orientation that globally obtains the best similarity score is denoted CRPP and is selected for further calculation.

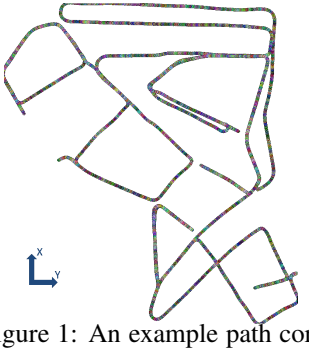


Figure 1: An example path corresponding to a set of poses resulting from the SfM algorithm.

## 3.2 Cross-route pose pair matching

When a suitable CRPP is obtained, the effective correspondence search between the images linked to $p_{\tilde{r}_k}$ and the images linked to $p_{\tilde{r}_l}$ is initiated. We have the stages earlier introduced: single pose cloud construction, cross-route image matching and PROSAC robust estimation.

### 3.2.1 Single Pose Cloud Construction

The poses in the CRPP $= \left(p_{\tilde{r}_k}, p_{\tilde{r}_l}\right)$ are treated separately at first. In the following section, $p_{\tilde{r}}$ will be used as the pose in question.

Using the four stereo camera pairs that were introduced earlier, accurate positions of physical points surrounding $p_{\tilde{r}}$ are extracted. The downscaled images are subjected to interest points detection and description, again using the SURF scheme [1]. The resulting descriptors for every stereo pair are matched: $I_{\tilde{r},j} \leftrightarrow I_{\tilde{r},j+1}$, for $j = 0,2,4,6$.

[1] $|X|$ denotes the cardinality or size of the set $X$.

[2] The number of calculations is reduced by a factor 4 when only considering one image of each stereo pair instead of going through all cameras. The increase in accuracy is minimal, since the overlap between the images of a stereo pair is substantial.

Three pruning methods are built in. First of all, *crossmatching* is performed as in the NB framework, whereby it is demanded that a match from left to right image is consistent with the corresponding match from right to left image.

Another pruning is built in based on the Nearest Neighbor Distance Ratio (NNDR) method as in [9, 10]. This condition translates the demand for a certain distinctiveness of the keypoint descriptors and easily rules out ambiguities that can occur in repetitive patterns for example. Here, we fix the NNDR threshold to 0.8.

Finally, another intervention is pursued to increase the confidence in the tentative matches. Since the calibration parameters of the cameras are available, an epipolar constraint is imposed. To relate two cameras that are close, the fundamental matrix $F$ is used. For a pair of proposed corresponding points $\mathbf{x}_{c_i}$ and $\mathbf{x}_{c_{i+1}}$ the following must thus hold (with $\varepsilon \approx 0.01$): $\mathbf{x}_{j,c_{i+1}}^{\top} F_{c_i,c_{i+1}} \mathbf{x}_{j,c_i} < \varepsilon$, where $F_{c_i,c_{i+1}}$ is directly found from the cameras' projection matrices.

Triangulation of the surviving correspondences is straightforward, using the camera calibration and the algorithm provided in [8]. The result is a point cloud of physical points that denote distinctive elements of buildings, street features and other urban elements, originating from 8 images and 4 matching procedures in total. Only four stereo pairs are available, and only the regions that are contained in both fields of view of a stereo pair can result in triangulation. The result is a point cloud with irregular occupancy around the van.

To augment the point cloud with even more points, one can resort to the previous and the next poses of the van on the same route. Again using the three tests, matches are tracked between images $I_{\tilde{r},j}$ of the current pose $p_{\tilde{r}}$ and camera $c_j$ and $I_{\tilde{r}+1,j}$ of the next pose $p_{\tilde{r}+1}$ and the same camera $c_j$. Similarly, this is done between $I_{\tilde{r},j}$ and $I_{\tilde{r}-1,j}$ of the previous pose in the same route. Next pose and previous pose matching are each carried out for four images, *i.e.* one image for each stereo pair:

$$I_{\tilde{r},j} \leftrightarrow I_{\tilde{r}+1,j}, \quad I_{\tilde{r},j} \leftrightarrow I_{\tilde{r}-1,j}, \text{ for } j = 0,2,4,6 \tag{1}$$

The epipolar check can still be carried out, since the drift error in translation and orientation between two subsequent poses is negligible. The result of this extra matching step is a more uniform point density of the cloud around the van. The computation of such a denser cloud however is time expensive and should only be calculated when needed, *i.e.* when a transformation between two clouds (see 3.2.2) was not found.

### 3.2.2 Cross-Pose Image Matching

Now that there is geometrical information available for $p_{\tilde{r}_k}$ and $p_{\tilde{r}_l}$ separately, the two images from the different routes, are now matched to each other:

$$I_{\tilde{r}_k,j} \quad \leftrightarrow \quad I_{\tilde{r}_l,j'}, \text{ for } j = 0,2,4,6. \tag{2}$$

where $j'$ is determined based on the NB framework results and now informs the systems on the image combinations that should be matched. Since this orientation led to a minimal score in the NB scheme, it is safe to assume that the matching should follow this line of work. In fact, the NB scheme used a very coarse, fast but exhaustive cross-route image matching scheme, which is now done carefully.

The matching is again performed by using the SURF descriptors. In this point of the algorithm, no further feature extraction is needed since this has already been done to construct a cloud around each pose. To be able to relate image features to 3D points, only these features are retained that actually led to a triangulated point during the cloud construction. These features previously survived the three tests, proving that they were distinctive on a

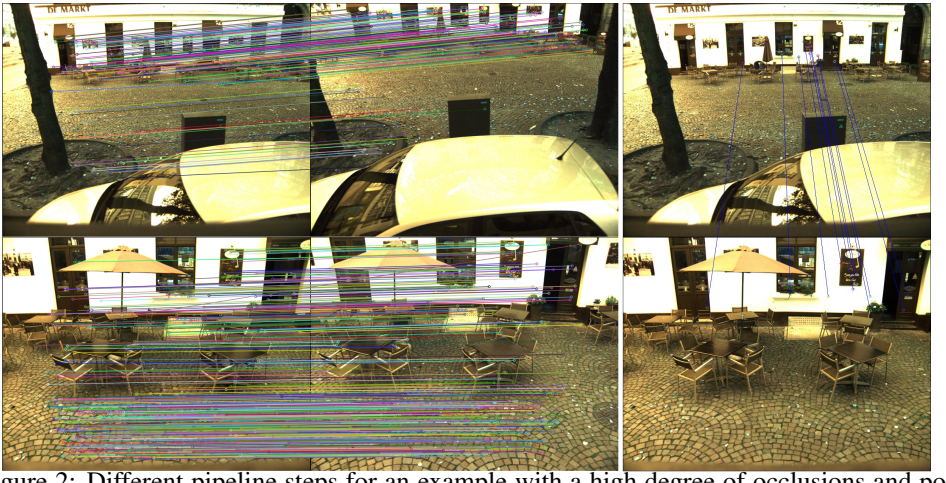


Figure 2: Different pipeline steps for an example with a high degree of occlusions and position discrepancy. Left: within correspondences for two poses after the three tests. Right: cross-pose image matches after PROSAC. Note that these are not the only correspondences found; also for other image combinations matches are tracked.

*same-route basis*. Matching the features that have 3D information associated, on a *cross-route basis* is performed by means of the crossmatching test as before as well as the NNDR test. In this stage, using the epipolar constraint does not make sense, since the purpose of the entire algorithm is to detect inconsistencies between poses that relate to very distant recordings. We cannot trust on the geometric relations between cameras linked to two such poses and we compute arrays of putative *cross-route image correspondences* (CRIC) containing 3D information results. It should be noted very clearly that the resulting CRICs are a collection of the 4 matching algorithms defined in Equation (2). The typical view-view approach is extended to a pose-pose approach.

### 3.2.3 PROSAC robust estimation

For the remaining putative CRICs, that represents putative links between the two single pose point clouds, a RANSAC scheme is run [5] that removes false connections (see Figure 2). Since scores are available from the NNDR test, a prioritized RANSAC scheme can be devised. We use the Progressive Sample Consensus (PROSAC) [2] algorithm. The PROSAC draws samples from progressively larger sets of top-ranked correspondences and typically a speed-up of two order of magnitude is expected as stated in [2] with respect to RANSAC.

If a transformation is found, its inliers are denoted true cross-route correspondences and all information that lead to the determination of the 3D points are reintroduced in the bundle adjustment system. Accuracy is expected to increase substantially by adding this information. The earlier remark that false positives are to be avoided at all cost is respected due to the continuous pruning out of bad results and the RANSAC scheme.

## 4 Experimental Setup & Results

**Dataset & Test Bench.** Since it concerns a system-specific application, a specialized dataset is devised that comprises a substantial amount of images from an urban environment. Two different recordings were tested. The images are $618 \times 814$ pixels. The first dataset, `GRB_03_4`

| # | #DNB | Descriptor | INC | Matching Rate | | Comp. Time | |
|---|---|---|---|---|---|---|---|
| 1 | 100 | SURF-128 | ✗ | 44/49 | 89.8 % | 1232 s | 100.0% |
| 2 | 100 | SURF-128 | ✓ | 45/49 | 91.8 % | 887 s | 72.0% |
| 3 | 20 | SURF-128 | ✗ | 40/49 | 81.6 % | 1114 s | 90.4% |
| 4 | 20 | SURF-128 | ✓ | 43/49 | 87.8 % | 746 s | 60.6% |
| 5 | 100 | SURF-64 | ✗ | 46/49 | 93.9 % | 1312 s | 106.5% |
| 6 | 100 | SURF-64 | ✓ | 48/49 | 98.0 % | 782 s | 63.5% |
| 7 | 20 | SURF-64 | ✗ | 45/49 | 91.8 % | 1224 s | 99.4% |
| 8 | 20 | SURF-64 | ✓ | 45/49 | 91.8 % | 624 s | 50.6% |
| 9 | 100 | USURF-128 | ✗ | 45/49 | 91.8 % | 1188 s | 96.4% |
| 10 | 100 | USURF-128 | ✓ | 46/49 | 93.9 % | 724 s | 58.8% |
| 11 | 20 | USURF-128 | ✗ | 46/49 | 93.9 % | 1073 s | 87.1% |
| 12 | 20 | USURF-128 | ✓ | 46/49 | 93.9 % | 568 s | 46.1% |
| 13 | 100 | USURF-64 | ✗ | 48/49 | 98.0 % | 1259 s | 102.2% |
| 14 | 100 | USURF-64 | ✓ | 48/49 | 98.0 % | 592 s | 48.1% |
| 15 | 20 | USURF-64 | ✗ | 47/49 | 95.9 % | 1177 s | 95.5% |
| 16 | 20 | USURF-64 | ✓ | 47/49 | 95.9 % | 483 s | 39.2% |
| 17 | 10 | USURF-64 | ✓ | 42/49 | 85.7 % | 534 s | 43.4% |

Table 1: Investigation of time decreasing techniques on GRB_03_4 dataset. *#DNB* stands for number of descriptors used for NB matching. *Descriptor* provides type and dimension of the descriptor. *INC* shows if the incremental approach is enabled. *Matching rate* is given in ratios and percentages, *i.e.* the amount of stitched scenes out of the total number of extracted CRPPs. *Computation time* is provided in seconds, relative to the Setup 1, the baseline.

(5531 poses), was used to investigate the different time decreasing measures and tuning the method. The second one, GRB_02_1 (14999 poses), was used to test the method and to build up the mockup examples. Prior to calculations, the poses are subsampled to a subset with approximately evenly spaced (1m) poses. Experiments were carried out using a desktop computer equipped with an i5-3570 3.40Ghz processor.

**Time-decreasing techniques.** Table 1 summarizes the results that justify the time-decreasing techniques proposed throughout. Lowering the number of descriptors used for Naive Bayes Matching typically represents a 15% decrease in computation, while the drop in matching rate is not substantial, as shown for image classification in [22]. The use of shorter descriptor lengths and the simpler Upright SURF version[3] during the cloud construction and the *CRPP*-matching have a positive impact on timing as well as results. The incremental method that does not directly perform triangulation using $p_{prev}$ and $p_{next}$ in every route has a highly positive effect on computation time: cross-route image matches are already found when a sparse cloud of irregular density is queried. On top of this, detection rate is often even higher for the incremental approach, revealing that overloading the clouds with points can reduce the detection probability for the cross-pose image matching and subsequent RANSAC procedure.

Incremental USURF-64 using NB20, *i.e.* rotation-variant Upright SURF using 64 dimensional feature vectors and 20 descriptors in NB matching, proved to be a good basis for validation. To further research if this setup allows a further drop of the number of features for NB, setup 17 was tested, but the running time increased (more often a richer cloud had to be extracted) and the performance degraded as well.

To reveal where the most computation-demanding steps in the algorithm are located, a breakdown experiment was performed on two setups, namely setup 13 (top-performing yet time-expensive) and 16 (well-performing and fast). The results of these setups, on dataset GRB_03_4, are shown in Figure 3. Two different break-ups are treated, of which the second

[3]SURF where the rotation invariance is disabled. This is a wishful property, since the number of ambiguities decreases in this setting.

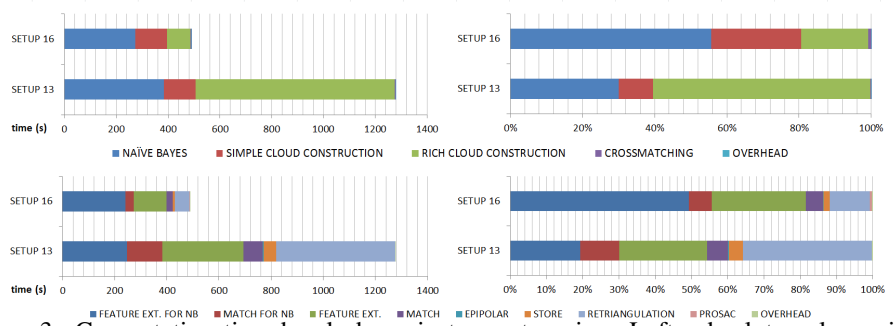



Figure 3: Computation time break-down in two categories. Left: absolute values, right: percentual values. The upper two bars denote the division in main steps. The lower bar goes further into detail, revealing the bottlenecks.

| # | #DNB | Descriptor | INC | ANC | Matching Rate | | Comp. Time | |
|---|---|---|---|---|---|---|---|---|
| 1 | 100 | USURF-64 | ✓ | ✗ | 213/234 | 91.0% | 2988 s | 100.0% |
| 2 | 20 | USURF-64 | ✓ | ✗ | 220/234 | 94.0% | 2307 s | 77.2% |
| 3 | 100 | USURF-64 | ✓ | ✓ | 61/72 | 84.7% | 904 s | 30.3% |
| 4 | 20 | USURF-64 | ✓ | ✓ | 55/64 | 85.9% | 695 s | 23.3% |
| 5 | 100E | USURF-64 | ✓ | ✗ | 226/234 | 96.6% | 3409 s | 114.0% |
| 6 | 20E | USURF-64 | ✓ | ✗ | 230/234 | 98.3% | 2563 s | 85.8% |
| 7 | 100E | USURF-64 | ✓ | ✓ | 61/65 | 93.8% | 1055 s | 35.3% |
| 8 | 20E | USURF-64 | ✓ | ✓ | 63/65 | 96.9% | 898 s | 30.1% |

Table 2: Validation of the determined setup on `GRB_02_1` dataset. We report as in Table 1, and for another time-decreasing method is introduced, namely the *anchoring*, denoted ANC. `E` in the #DNB property designates the exhaustive approach.

one is the most unraveling. It seems that image loading and feature computation, for the NB step as well as for later image description, and retriangulation (in non-incremental mode) form major bottlenecks; they account for 77% of the processing time for the fast Setup 16.

**Validation.** The determined setup is applied to a different, longer dataset (`GRB_02_1`) to confirm our conclusions. Specifically for the envisioned application, there is no need to have loop-closing information every several meters in order to greatly enhance accuracy. For this reason an extremely simple yet effective measure was introduced, that neglects the clusters following a detected cluster within a reasonable distance (in experiments set to 50m). The number of treated clusters reduces, and thus the overall computation time drops significantly. Results of this approach are added to previous measures, with the different setups summarized in Table 2. As an addition, to check whether matching rate will increase if several NB ranked combinations are checked if for the top scoring candidate *CRPP* no matching was found, 4 setups were added. We can conclude that the selected setup is justified, and that exhaustive matching, improves the matching rate but at the cost of increased running time.

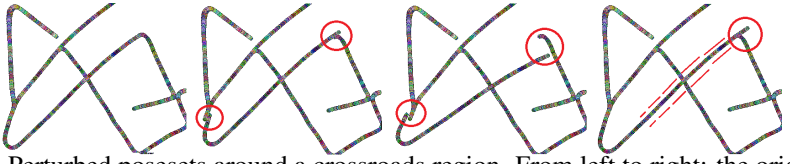
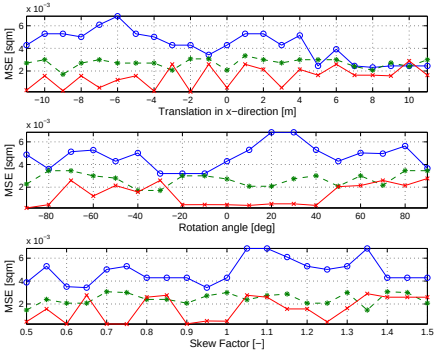
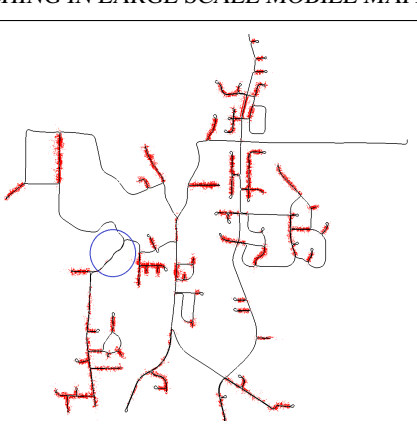


Figure 4: Perturbed posesets around a crossroads region. From left to right: the original excerpt, poseset perturbed by translation, by combined translation and rotation, and by skewing, where the translation is perturbed increasingly from a certain pose on, mimicking the effects of very substantial drift. The red markings show the endpoints for the perturbation.




(a) Robustness to perturbations in MSE terms on three representative mock-up examples: crossroads (*blue* line), same direction (*green*) and different directions (*red*).



(b) Example of system embedding, where all the matched points are visualized in red. The van's path is drawn in black.

Figure 5

**Mockup Examples.** The dataset that was worked with already has a high degree of accuracy already; the SfM algorithm did a good job without loop closing techniques. However, to fully test our algorithm and to prove that it is, apart from a very low GPS dependency, robust to many types of deformations, three excerpts from the database (crossroads, concatenating routes in both same and opposite direction) were perturbed, as demonstrated in Figure 4. Thus, for two overlapping routes/posesets (distant recordings, therefore with different images and estimations for their poses and camera parameters), only one poseset was artificially perturbed to mimic the SfM drift in the estimated parameters. For translations a shift in $x$-coordinate was applied, for the rotations the posesets were turned around an anchor pose and for the skew, the translations of the pose were perturbed to mimic a substantial drift. On the perturbed mockup our procedure was used to restore from the perturbation by finding the local transformation from one poseset to the other. Then we computed the errors between the original poseset before applying the perturbation and the restored poseset. The results (see Figure 5a) show that mean square error (MSE) is not dependent of the applied transformation. However, for the most challenging crossroads excerpt, the MSE is considerably larger. No systematic error is apparent from these tests.

**System embedding.** Figure 5b shows all resulting sets of CRICs for an entire recording in a suburban American area. One clearly sees that almost everywhere the van passed more than once the system was able to extract a high number of matches (marked in red). When routes cross, a limited number or no matches are found. The blue circle marks a site with many trees. The associated lack of distinctiveness for the features extracted from this site leads to few found correspondences. The integration of our loop closure results into the current bundle adjustment algorithm is expected to increase the accuracy of the system.

**Discussion.** Overall, it can be stated that our system is robust to noise (see Figure 5a), while achieving high matching rates (see Figure 2). The system outperforms state-of-the-art approaches, but uses the prior built map for quick candidate selection which thus demands careful comparison. Furthermore, as shown in Figure 3, the major bottleneck in the system is feature extraction, both in the NB matching step and in the image matching itself. Finally, the processing time is high, but the addition of this technique is expected to leverage a time decrease by two orders of magnitude compared to the original bundle adjustment of the system. The number of false positives was zero for all experiments.

# 5 Conclusion

In this paper, an original approach to the loop closing problem was proposed. By extending the typical image-to-image matching scheme to a general pose-to-pose matching technique, the matching rate showed to increase substantially in numerous experiments. Comparison with state-of-the-art techniques is however not straightforward because of differing datasets. Several time-decreasing steps were thoroughly researched in order to lighten the system requirements, where considerable decrease can be obtained without compromising performance drastically. Future work is to construct a confidence measure in order to inform the bundle adjustment system of the reliability and precision of the information. The feature extraction was found to be a bottleneck, and alternative techniques for description in the Bayesian framework need to be researched. Finally, further thorough validation of the developed technique must be carried out.

**Acknowledgments.** This work was partly supported by the ERC Advanced Grant VarCity and the EC FP7 Strep project ROVINA. We thank GeoAutomation for providing the data.

# References

[1] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 2008.

[2] Ondrej Chum and Jiri Matas. Matching with PROSAC - Progressive Sample Consensus. In *Proceedings of Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[3] Mark Cummins and Paul Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 2008.

[4] Mark Cummins and Paul Newman. Accelerated appearance-only SLAM. In *Proc. IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena,California, April 2008.

[5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

[6] A. Glover, W. Maddern, M. Warren, S. Reid, M. Milford, and G. Wyeth. Openfabmap: An open source toolbox for appearance-based loop closure detection. In *The International Conference on Robotics and Automation*, St Paul, Minnesota, 2011. IEEE.

[7] Chris Harris and Mike Stephens. A combined corner and edge detector. In *In Proc. of Fourth Alvey Vision Conference*, 1988.

[8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2000.

[9] Jing Huang and Suya You. Point cloud matching based on 3d self-similarity. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012.

[10] Tony Lindeberg. Feature detection with automatic scale selection. *Int. J. Comput. Vision*, 1998.

[11] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2), 2004.

[12] Will Maddern, Michael Milford, and Gordon Wyeth. CAT-SLAM: probabilistic localisation and mapping using a continuous appearance-based trajectory. *Int. J. Rob. Res.*, 2012.

[13] William P. Maddern, Michael Milford, and Gordon Wyeth. Towards persistent localization and mapping with a continuous appearance-based topology. In *Robotics: Science and Systems*, 2012.

[14] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *Int. J. Comput. Vision*, 2004.

[15] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application (VISSAPP)*, 2009.

[16] Rohan Paul and Paul Newman. FAB-MAP 3D: topological mapping with spatial and visual appearance. In *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2010.

[17] Klaus Peter Schwarz and Naser El-sheimy. Mobile mapping systems - state of the art and future trends. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35:10, 2004.

[18] Stephen Se, David Lowe, and Jim Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *The International Journal of Robotics Research*, 2002.

[19] C Silpa-Anan and R Hartley. Visual localization and loop-back detection with a high resolution omnidirectional camera. In *Proceedings of the tirteenth IEEE International Conference on Computer Vision (ICCV)*, 2005.

[20] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV)*, 2003.

[21] Niko Sünderhauf and Peter Protzel. Brief-gist-closing the loop by simple means. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, 2011.

[22] Radu Timofte, Tinne Tuytelaars, and Luc Van Gool. Naive bayes image classification: beyond nearest neighbors. In *Asian Conference on Computer Vision (ACCV)*, 2012.