LOGOS: Local geometric support for high-outlier spatial verification*

Stephanie Lowry and Henrik Andreasson[1]

***Abstract*— This paper presents LOGOS, a method of spatial verification for visual localization that is robust in the presence of a high proportion of outliers. LOGOS uses scale and orientation information from local neighbourhoods of features to determine which points are likely to be inliers. The inlier points can be used for secondary localization verification and pose estimation. LOGOS is demonstrated on a number of benchmark localization datasets and outperforms RANSAC as a method of outlier removal and localization verification in scenarios that require robustness to many outliers.**

I. INTRODUCTION

Robust visual localization is an important part of a visual navigation system. The goal of visual localization is to determine if the current observation is from the same location as other, previously-seen observations, and the first step of a visual localization system is often to compare a query image to a set of database images. For a robot operating in a large environment, the number of images to be matched may be extremely high, and the system needs to be efficient, both in memory requirements and computational requirements. Such restrictions can have a major effect on retrieval quality [1]. Thus it is common when performing visual localization that the efficient and large-scale initial matching process is refined in a secondary step which re-evaluates and re-ranks the candidates proposed by the initial process.

A common way to verify a match is to use spatial information about the scene. The correspondence between views of the same scene is well-known [2], and can be used to assess using vision geometry whether two images represent two different views of the same scene. However, such spatial verification systems can fail in the present of a large number of outliers, and as the amount of memory available to be stored for each location is reduced, a greater number of outliers occur.

The contribution of this paper is a novel method of verification that succeeds in the presence of a high proportion of outliers (see Fig. 1). It introduces the concept of local geometric support where orientation and scale feature data from local neighbourhoods are used to determine which features are likely to be inliers. The verification system LOGOS significantly increased recall at 100% precision on all tested datasets, and in a test with 90% outliers, LOGOS found 31% of inliers while RANSAC found fewer than 1%. LOGOS is model-free, and thus does not guarantee the inliers satisfy a concrete transformation or epipolar geometry, but this paper demonstrates that the inliers identified by LOGOS can also be used for pose estimation if 3D point information is available.

*This work is supported by the Semantic Robots Research Profile, funded by the Swedish Knowledge Foundation (KKS).

[1]The authors are with the Centre for Applied Autonomous Sensor Systems (AASS), Örebro University, 70281 Örebro, Sweden. Email: stephanie.lowry@oru.se

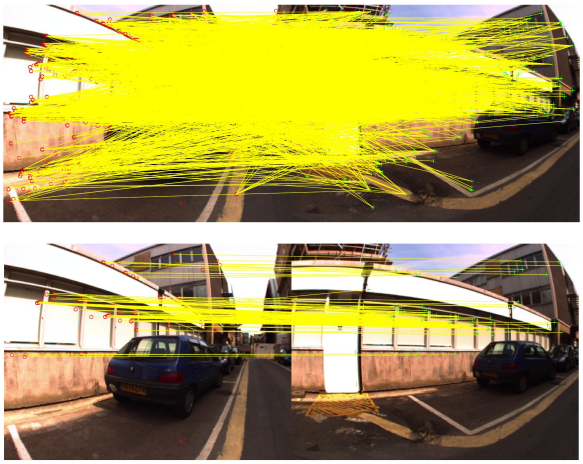


Fig. 1. LOGOS selects an inlier set of point matches for spatial verification of location matches using orientation and scale information from local neighbourhoods of points. In this example, from the Oxford City Centre dataset, LOGOS finds 110 inlier points from a original set of 2604, eliminating approximately 95% of points as outliers.

II. PRIOR WORK

Spatial verification of scenes to determine place matches frequently uses a variant of RANdom SAmple Consensus (RANSAC) [3] to estimate the parameters of the model. RANSAC has a number of appealing characteristics: its computation time is largely independent of the number of image points and the model it calculates is not influenced by the presence of outliers. Enhancements of the basic RANSAC algorithm include MSAC [4], LO-RANSAC, [5], and PROSAC [6].

RANSAC and its variants typically cannot be used when the proportion of outliers is high. An exception is [7], which considers the distribution of residuals for all hypotheses for individual data points to determine which data points agree with a large number of hypotheses and are therefore likely to be inliers, and demonstrates inlier identification when outliers form up to 70% of the data.

Alternatively, spatial verification can be performed using geometric support based on feature information such as orientation and scale. Feature information can be combined with a Hough Transform to predict an affine transformation between objects in a scene [8], and is robust to very high proportions of outliers (over 95%). Other methods do not attempt to compute the transformation, and instead perform

a consistency check over a global scale [9], [10]. In [11], the global consistency check is followed by a pairwise comparison routine that weights each geometric relation by its agreement with other correspondences.

If scene image matching is occurring under large viewpoint change, the overlapping regions between the two images may be small. [12] propose a 2-point RANSAC process that assumes that camera motion is planar and most features are grouped into vertical planes and reject outliers via multiple homography matrices.

One key novelty of this work is to propose the concept of *local geometric support*, where geometric support is calculated in local neighbourhoods to select inliers. Such a method is extremely robust to high outlier ratios, without making strong assumptions about the environment structure or camera motion.

III. APPROACH

The goal of LOGOS is to determine a set of inlier points from a pair of candidate image matches (see Fig. 1). When there are many outliers, outlier rejection techniques that rely on global information will fail. However, LOGOS uses local information about a point to decide whether it is an outlier. Assuming a typical real-world environment, points that are generated by objects close to each other in 3D space, will transform (move, rotate, and scale) in a roughly similar way. Thus if two points nearby to each other within the image transform in a similar way, they provide local geometric support for each other and can be considered as inliers.

A single similar transformation (such as two points rotating in the same way) does not on its own provide a compelling argument for geometric support, but multiple different rotation and scaling consistency checks together build up a body of evidence that is highly unlikely to have occurred randomly. Furthermore, since multiple inlier points need to be found in an image for it to be considered a likely inlier set, an occasional false support point is unlikely to cause overall system failure. However, LOGOS does depend on feature points being observed in the environment with sufficient density that local geometric support points can be found.

LOGOS exploits the orientation and scale information provided by many local feature detectors such as SIFT [8], SURF [13], and BRISK [14]. This section defines the necessary requirements for calculating local geometric support and determining inliers for an image pair (see also Fig. 2). In the following discussion, for each point p, we define the 2D location of p within the image as $x_p = (x, y)$, the orientation of p as θ_p and the scale of p as s_p.

A. Nearest neighbour selection

Each image point p in image I can receive local geometric support from its neighbourhood points. The neighbourhood of p, denoted $\mathcal{N}$, is defined as the set of N image points in I that have the closest Euclidean distance to p.

The closest points within I are not always the true closest points within the real 3D environment. Consistency checks are undertaken between p and the points in $\mathcal{N}$ to remove false nearest neighbours, but for LOGOS to successfully identify p as an inlier, some of p's true nearest neighbours must be included in $\mathcal{N}$. The choice of the neighbourhood size parameter N must balance between minimising false nearest neighbours and ensuring true support points are included.

B. Identify local support points

Let (I, I') be a potential candidate pair of image matches and (p, p') be a possible point match between I and I', with $\mathcal{N}$ and $\mathcal{N}'$ the nearest neighbour sets of p and p' respectively. If there is another possible point match q and q' such that $q \in \mathcal{N}$ and $q' \in \mathcal{N}'$ then q is a potential support point of p. If q also satisfies the orientation and scale checks described in Sections III-C and III-D below, it will be accepted as a true support point of p.

C. Intra-orientation and intra-scale consistency

The first consistency checks ensure that the relative internal orientations and scales of p and q agree (see Fig. 2a). Define the relative orientations $\tilde{\theta}_p = \theta_p - \theta_{p'}$ and $\tilde{\theta}_q = \theta_q - \theta_{q'}$. For q to be a support point of p, these relative orientations must agree with each other to within a stated accuracy Θ_{intra}:

$$| \tilde{\theta}_p - \tilde{\theta}_q | < \Theta_{\text{intra}} \tag{1}$$

Similarly the relative scales should agree. Since feature scale is best considered as a ratio, LOGOS compares relative log-scales $\tilde{s}_p = \log(s_p) - \log(s_{p'})$ and $\tilde{s}_q = \log(s_q) - \log(s_{q'})$:

$$| \tilde{s}_p - \tilde{s}_q | < S_{\text{intra}} \tag{2}$$

D. Inter-orientation and inter-scale consistency

These consistency checks ensure that the relative orientation and scale of the vectors between the points agree with the internal values (see Fig. 2b). Define vector $x = x_p - x_q$ in image I and vector $x' = x_{p'} - x_{q'}$ in image I'. The inter-point orientation and scale relations are defined as in [11] by

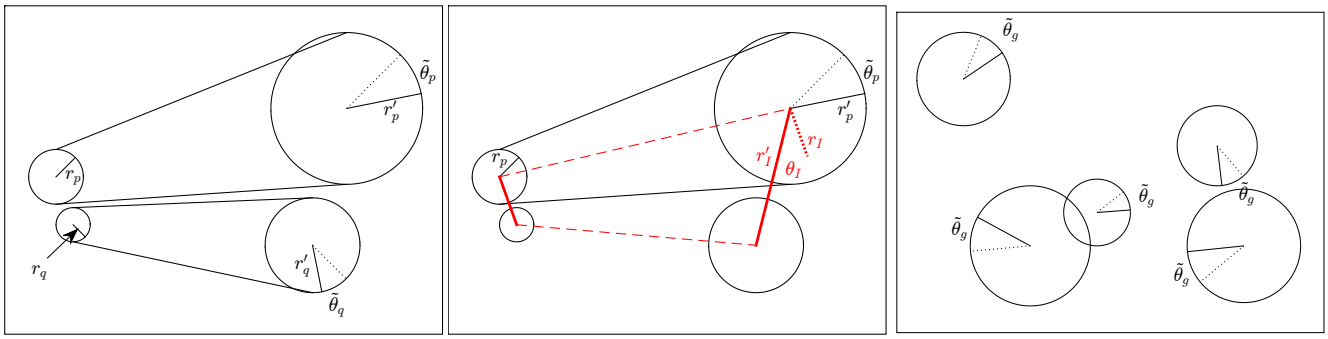
$$\theta_I = \arccos\left(\frac{x \cdot x'}{||x||||x'||}\right) \cdot \operatorname{sgn}(x \times x') \tag{3}$$

$$s_I = \log(||x'||) - \log(||x||) \tag{4}$$

These external orientation and scale values must match internal orientation and scale values $\tilde{\theta}_p$ and $\tilde{s_p}$:

$$| \theta_I - \tilde{\theta}_p | < \Theta_{\text{inter}} \tag{5}$$

$$| s_I - \tilde{s}_p | < S_{\text{inter}} \tag{6}$$



(a) Intra-point consistency (b) Inter-point consistency (c) Global consistency

Fig. 2. LOGOS consistency checks. The black circles denote image points, with the size of the circle representing scale and the black lines within the circles denoting orientation. (a) Intra-point orientation and scale consistency check. The two points on the left-hand side transform to the points on the right-hand side with similar relative orientations ($\tilde{\theta}_p$ and $\tilde{\theta}_q$) and scale (illustrated by the ratio between r_p and r'_p and r_q and r'_q). (b) Inter-point orientation and scale consistency check. The solid red lines are the vectors between the two point centres and the relative orientation (θ_I) and scale (illustrated as the ratio between r_I and r'_I) of these vectors must match the internal point orientation $\tilde{\theta}_p$ and scale (ratio between r_p and r'_p). (c) Global consistency check. All inlier points must have the same relative orientation θ_g.

E. Global consistency check

If a local support point q is found for point p, (p, p') is added to the set $\mathcal{P}_\mathrm{L}$ of point matches that are deemed to possess local geometric support. The results are further refined on a global scale so that all inlier points share a relative orientation (see Fig. 2c). The global relative orientation $\tilde{\theta}_\mathrm{g}$ is determined from the relative orientations of all the pairs in $\mathcal{P}_\mathrm{L}$, which are grouped into bins of size $\frac{\Theta_\mathrm{global}}{3}$, where Θ_global is the global orientation cutoff limit. $\tilde{\theta}_\mathrm{g}$ is calculated as the most common orientation based on a sliding window of 3 bins, and any points that do not satisfy the global consistency check are omitted:

$$| \tilde{\theta}_\mathrm{g} - \tilde{\theta}_p | < \Theta_\mathrm{global} \tag{7}$$

The remaining points $\mathcal{P}$ that satisfy Eqn. 7 are the final inliers of the image pair (I, I'). The image pair can be accepted as a correct or incorrect match according to the number of inliers.

F. Parameters

Each of the consistency checks has a cutoff limit that determines whether or not it is considered to be a match. Table I summarizes the key parameters for LOGOS.

IV. EXPERIMENTAL SETUP

These experiments compare LOGOS[1] against RANSAC, the canonical technique for both spatial verification for place recognition and pose estimation. .

A. System workflow

Fig. 3 presents a schematic of the system workflow. A query image is compared against a database of previously seen images to select the most likely match. Often the top-k candidates are selected, but for simplicity these experiments use only the single most likely match. Potential inlier matches are determined between these images and then the spatial verification system (LOGOS or RANSAC) makes a final decision as to whether accept or reject the suggested match.

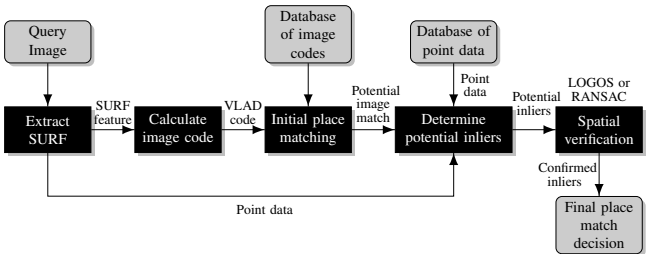


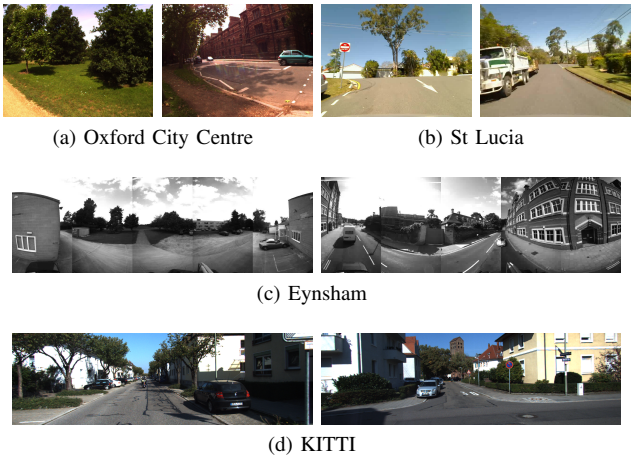
Fig. 3. Workflow diagram of localization system. The query image is compared against a database of previously seen images to determine potential matches. These potential matches are then inspected by the spatial verification system to make a final decision to accept or reject the suggested match.

1) Place matching method: Vectors of Locally Aggregated Descriptors (VLAD) [15] are used to identify potential location matches. VLAD can match places using very small image codes, and so is suitable for a low-memory visual localization application. VLAD is based on a bag-of-words model that partitions the feature space of into k visual "words" and assigns each image feature to a particular word. This work uses a 512-word model for VLAD and uses 128-dimension SURF features as the underlying feature detector and descriptor.

2) Determining potential inliers: Each SURF feature extracted from an image is used to calculate the resulting image code. Each feature is associated to a particular word in the VLAD model, and potential inliers are defined as follows. If database image I_d is the most likely location match for query image I_q, and feature f in I_q is assigned to word w, then for any feature f' in I_d also assigned to word w the pair (f, f') is a potential inlier pair.

3) Spatial verification method: The potential inliers determined above are used as input to the spatial verification

[1]LOGOS implementations in Matlab and C++ are available for download at `https://github.com/short-circuitt/LOGOS`



(a) Oxford City Centre (b) St Lucia

(c) Eynsham

(d) KITTI

Fig. 4. Sample images from (a) Oxford City Centre, (b) St Lucia, (c) Eynsham, and (d) KITTI datasets.

methods (LOGOS or RANSAC). The output of the spatial verification methods is a count of the number of inliers, and this count was used to decide whether or not the image pair (I_q, I_d) was a true location match.

B. Datasets

For this experiment four benchmark datasets were used, each of which provided a different test environment (see Fig. 4 for sample images). The Oxford City Centre dataset [16] consists of 2474 images collected by a robot along a 2km urban loop. The Eynsham dataset [17] consists of 9575 omnidirectional images captured using a Point Gray Ladybug2 at approximately every 4 metres along 70km of urban, rural, and motorway roads. The images were captured at 1920×512 resolution, and to remove the repetitive road pixels these experiments only used the top 350 pixels from each image. The St Lucia dataset [18] consists of a 15km loop through suburban streets in Australia. For this work images were sampled from the 12:10pm dataset at approximately 3Hz for a total of 3850 images. Finally, the KITTI dataset [19] is a car-based dataset captured in Karlsruhe. We used the '00' sequence from the visual odometry suite, which contains 4541 stereo rectified images. The localization step used the left-hand images, with the right-hand images also used for stereo pose estimation.

Each dataset contains ground truth data. The City Centre and Eynsham datasets each provide a binary ground truth – hand-labeled for the City Centre and based on GPS with a 40m tolerance for Eynsham. The KITTI dataset was considered to have a ground truth match if two matching places were found within 10m. The St Lucia dataset contains a less accurate GPS signal so a tolerance of 30m was used.

C. Parameters

Table I shows the key parameters for LOGOS and RANSAC. The RANSAC method used was 8-point MSAC (M-estimator SAmple and Consensus) [4], a variant that maximizes the likelihood of the chosen solution. The LOGOS parameters were selected to demonstrate performance

TABLE I

VERIFICATION PARAMETERS.

Symbol	Name	Default value	Eynsham value
LOGOS			
N	Neighbourhood size	5	5
Θ_{intra}	Intra-orientation limit	0.1	0.1
S_{intra}	Intra-scale limit	0.2	0.1
Θ_{inter}	Inter-orientation limit	0.2	0.1
S_{inter}	Inter-scale limit	0.2	0.05
Θ_{global}	Global orientation limit	0.2	0.05
RANSAC			
	Maximum number of trials	500	500
	Distance threshold	0.01	0.01

with a moderate choice of parameters – neither extremely strict or extremely weak – and a more in-depth survey of parameter effects was undertaken in Section V-C. The parameters were kept the same across all the tested datasets with the exception of the Eynsham dataset, which contained more open, unstructured views and required stricter cutoff limits, particularly for orientation, at the inter-point and global levels.

As the Eynsham images are five separate images combined into a single omnidirectional image, the standard RANSAC calculation was performed separately on each of the five images, and the number of inliers from each image was summed together.

D. Evaluation

A spatial verification method is intended to provide high precision localization, and thus a key metric in these experiments is the recall at 100% precision. Recall and precision are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Here TP is the number of true positive matches, FP is the number of false positive matches and FN is the number of false negative matches. The recall at 100% precision calculates what proportion of true positives can be identified without erroneously including any false positive. As the recall at 100% precision is highly sensitive and can be affected by a single false positive image, some experiments also quote the recall at 99% precision. This metric is of less practical importance, but is less sensitive to noise and so can provide additional information.

V. RESULTS

A. Visual localization

The first experiment compared the performance of LOGOS and RANSAC as spatial verification methods on the four datasets described in Section IV-B above. For each image, the best matching image was determined using VLAD. The spatial verification systems re-evaluated and re-ranked

these matches according to the number of inliers found, and the recall at 100% precision was calculated. Recall at 100% precision was also computed for the initial place-matching method based on the distance between the VLAD descriptors, without the secondary verification step.

Fig. 5 shows the recall at 100% precision on each of the tested datasets. The initial place matching method (black bars) achieves 0% on the Eynsham dataset and 3% on the City Centre dataset, while LOGOS (red bars) achieves 30% and 26% respectively. On the St Lucia and KITTI datasets LOGOS achieves 83% and 91% recall at 100% precision. RANSAC achieves 0% recall at 100% precision on all the tested datasets.

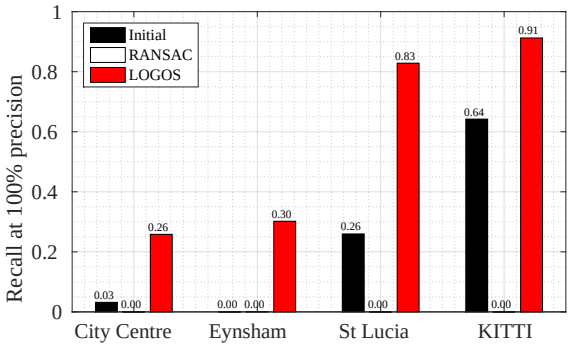


Fig. 5. Recall at 100% precision using the initial place match ranking, RANSAC and LOGOS. RANSAC achieves 0% recall at 100% precision for all datasets, while LOGOS outperforms the original ranking scheme in each case.

To illustrate some of the failure modes of LOGOS, Fig. 6a displays the three false matches from the City Centre dataset with the highest inlier counts (between 27 and 30), while Fig. 6b displays three true matches that also have an inlier count of 30. For each false match, LOGOS incorrectly matches small, similar-looking regions; the City Centre dataset is particularly challenging for it as it contains many frequently-repeated features such as window panes [16]. These failures, compared to the consistent global transformations exhibited by the inliers of the true matches, suggest that LOGOS could benefit from the additional of further tests that enforce coherency at a global scale.

B. Outlier study

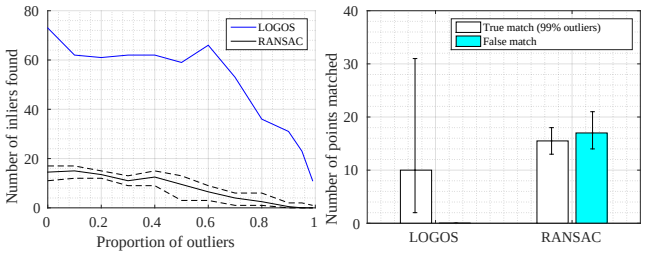
This section investigated the relationship between outlier numbers and spatial verification performance for LOGOS and RANSAC, using synthetic data to control the exact proportion of inliers and outliers. Images from the City Centre dataset were used, and 100 features were extracted from each. To ensure a known number of inliers, the identity transformation was used – that is, each feature mapped to itself and was a guaranteed inlier. However, random Gaussian noise was added to each point's location, scale and orientation (see Table II for noise parameters).

Synthetic outliers were added in. Position and orientation of the outliers was randomly selected from a uniform distribution (constrained to the size of the image for the position and to $[0, 2\pi)$ for the orientation), while to ensure a realistic distribution of scale values, the scale of the outlier points was drawn from a random sample of the scales of the inliers. To compensate for RANSAC's non-determininstic nature, RANSAC was run 10 times and the median, minimum, and maximum results were shown.

TABLE II
GAUSSIAN NOISE STANDARD DEVIATIONS.

Variable	Standard deviation
x position	2 pixels
y position	2 pixels
θ orientation	0.15 rad
s scale	0.1

1) Outlier proportion: Fig. 7a presents the number of inliers found by each method against the proportion of outliers on a single image. Overall, LOGOS finds a larger number of inliers than RANSAC. With 90% outliers, LOGOS finds 31 of the 100 inliers, while RANSAC finds 2 or fewer inliers on each sampled run. With 99% outliers, LOGOS finds 11 inliers while RANSAC finds 1 or fewer.



(a) (b)

Fig. 7. Synthetic experiment comparing the performance of LOGOS against RANSAC in the presence of outliers. (a) LOGOS outperforms RANSAC at detecting inliers, most crucially when there are many outliers (up to 99%). (b) Number of points identified as inliers by LOGOS and RANSAC. LOGOS identifies between 2 and 31 inliers for each true match, and 0 inliers for each false match, while RANSAC identifies between 13 and 18 inliers for each true match and between 14 and 21 inliers for each false match. RANSAC identifies a similar number of points in both true and false matches, and so cannot differentiate between true and false matches as effectively as LOGOS.

2) True matches with many outliers versus false matches: Spatial verification methods frequently decide which location matches are correct and which are incorrect based on the number of points identified as inliers. It is therefore important that inlier counts are widely different for true and false matches. In this experiment, ten images from the City Centre dataset were used, each with 10000 points – 100 inliers and 9900 synthetic outliers for the true matches and 10000 synthetic outliers for the false matches.

Fig. 7b shows the inlier count found for true matches with 99% outliers and for false matches. LOGOS find a median inlier count of 12 for the true matches (range from 2 to 31). For each of the false matches, LOGOS finds 0 inliers. In contrast, RANSAC finds an median inlier count of 15.5 for the true matches (range of 13 to 18) and a median inlier count of 17 for the false matches (range of 14 to 21). As

(a) Incorrect matches

(b) Correct matches

Fig. 6. Failure and success examples from the City Centre dataset. (a) The three highest-ranked false matches, with 30, 29, and 27 inlier matches respectively. In each case LOGOS erroneously matches small window features that are similar-looking and highly repetitive. (b) Three correct matches, each with 30 inlier matches. Compared to the incorrect matches, the inliers in each image display a coherent transformation.

RANSAC matches approximately the same number of points in each case, its inlier count cannot be used for differentiating between true matches with many outliers and completely false matches. In contrast, LOGOS inlier counts for true and false matches do not overlap.

C. Parameter study

LOGOS depends on a number of key parameters, and this experiment investigates the sensitivity of LOGOS to the choice of parameters. A range of parameter values were tested on the City Centre dataset using 200 images for evaluating the neighbourhood size and 270 images for evaluating the cutoff limits.

1) Nearest neighbour: Fig. 8 shows how different neighbourhood sizes affect LOGOS recall at 100% precision. Both too few and too many nearest neighbours impact performance. When only one nearest neighbour is allowed, there is the possibility that it may not be successfully identified in both images, and a true inlier will not receive support. When many nearest neighbours are permitted, the recall at 100% precision decreases, likely due to an increase in false support matches. The recall at 99% precision is less affected by a larger neighbourhood size, suggesting that false support matches are rare, but occur sufficiently often to interfere with the more sensitive matching at 100% precision. A compromise value of around 2 to 5 nearest neighbours works best.

2) Cutoff parameters: The cutoff parameters were evaluated across values from 0.05 to 1. For each data point, that parameter was held constant and all the other parameters were varied across the parameter space. Thus the results presented here show the best recall at 100% precision achieved for each parameter value.

Fig. 9 shows the relationship between recall and the cutoff limit parameters. Fig. 9a shows the orientation tests perform best with a cutoff limit of 0.1, where a recall of 70% at 100% precision is achieved on this data. The intra-orientation test is least sensitive to the choice of parameter, while the global orientation test is the most sensitive, decreasing to

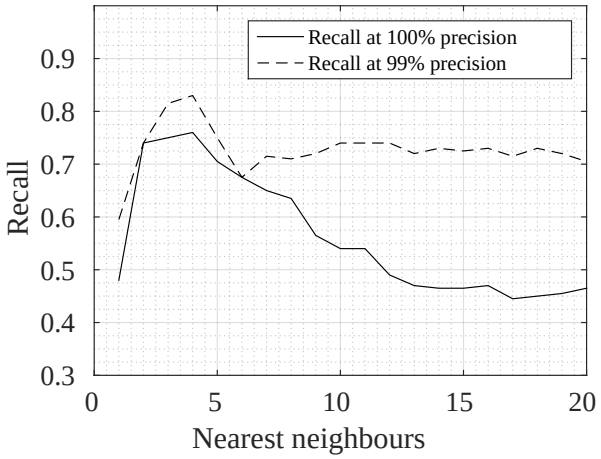


Fig. 8. Recall by LOGOS for different neighbourhood sizes. A small neighbourhood of 2 to 5 points provides the best performance.

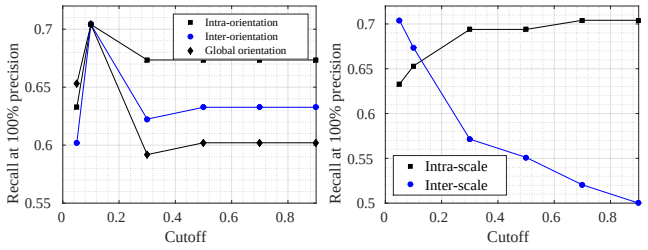
less than 60% recall at 100% precision for a cutoff of 0.3. Although optimal performance is sensitive to the choice of cutoff parameters, the performance is quite stable across a wide range of parameter values, changing very little when the parameters are varied between 0.3 and 1.

Fig. 9b shows that the choice of inter-scale parameter can be significant, with recall at 100% precision decreasing steadily as the cutoff limit increases. In contrast, recall at 100% increases as the intra-scale limit increases. This suggests that the intra-scale comparison may be more sensitive to noise, and thus a wider cutoff limit is necessary.

D. Memory and computation requirements

The memory requirements for LOGOS depend linearly on the number of feature points stored per image. This experiment investigates the relationship between feature number and LOGOS performance, as well as considering the required computation time.

Table III displays the memory requirements for LOGOS and RANSAC. Both methods require each feature to have a 2D position in the image (requiring 4 bytes per dimension



(a) Orientation (b) Scale

Fig. 9. Maximum recall at 100% precision for different values of LOGOS cutoff parameters for (a) orientation and (b) scale. A cutoff value of 0.1 is best for all the orientation tests. The scale limits have varying effects, as performance decreases as the inter-scale limit increases but increases as the intra-scale limit increases.

TABLE III

MEMORY REQUIREMENTS PER IMAGE.

Method	Number of features		
	100	500	1000
LOGOS	2kB	9kB	18kB
RANSAC	1kB	5kB	10kB
RANSAC (full descriptors)	51kB	254kB	508kB

or 8 bytes total) and a word label (2 bytes). LOGOS also requires orientation and scale information (8 bytes). In total, RANSAC requires 10 bytes per feature and LOGOS requires 18 bytes per feature. However, the previous experiments showed that RANSAC fails with such small memory requirements. If the full SURF descriptors are included to reduce the number of outliers, RANSAC then requires a total of 128×4 bytes for the descriptor plus 8 bytes for the position, or a total of 520 bytes per feature.

The amount of memory required depends linearly on the number of features stored per image. Fig. 10 shows the number of features also affects the solution quality. LOGOS achieves a much higher recall at 100% precision when 1000 features are stored, compared to only 100 features. In other words, a trade-off may be necessary depending on the amount of memory that can be stored per place and the importance of a very high level of precision.

Fig. 11 shows the relationship between feature number and computation time for LOGOS and RANSAC. As RANSAC uses a sampling method, it has a near-constant median computation time for all feature numbers. The computation time for LOGOS is dependent on the number of features – while the median computation time appears to flatten off, this is due to the nature of the available data as many images do not contain 1000 detected features. If only images with between 900 and 1000 features are evaluated, the median computation time is 0.2s (90th percentile: 0.3s). However, looking at the median time for LOGOS over the whole data sample, it is almost an order of magnitude faster than RANSAC (0.04s compared to 0.27s).

E. Robot pose estimation

If 3D depth information is available, a metric pose estimate can be calculated from an inlier set. A proof-of-concept study

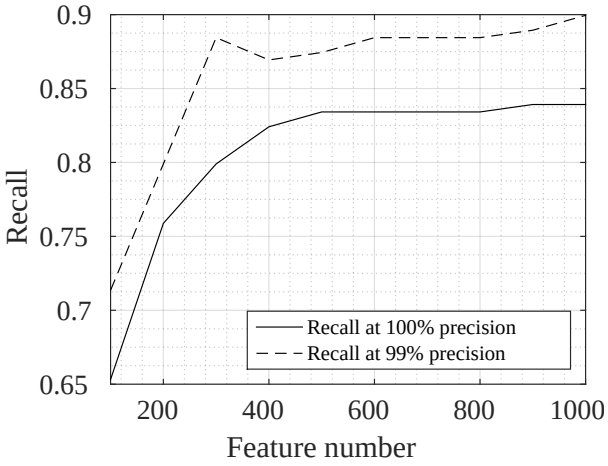


Fig. 10. Recall against feature number per image for LOGOS on the City Centre dataset. LOGOS achieves considerably higher recall when more features are stored; however, more memory is required to store the additional data.

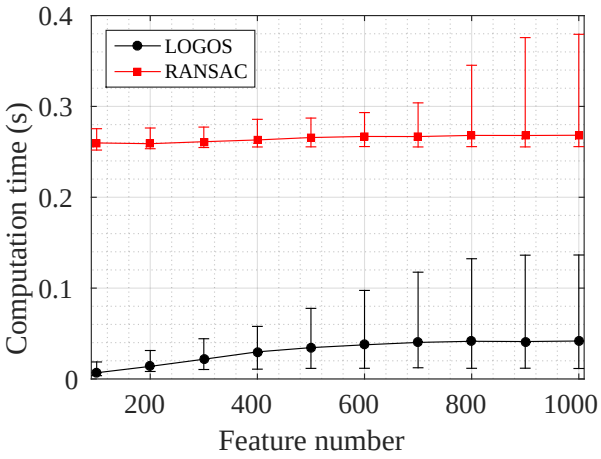


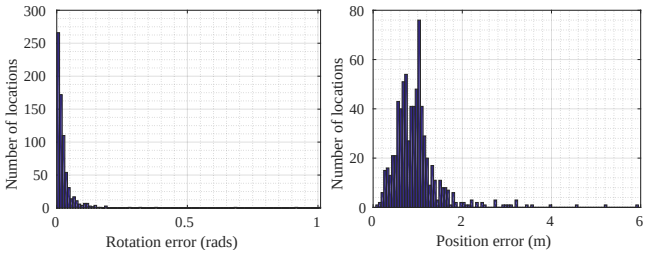
Fig. 11. Computation time compared to feature number for LOGOS and RANSAC. The line shows the median time per computation and the vertical bars show the 10th to 90th percentiles. The median computation time for RANSAC is nearly constant relative to feature number, while the computation time for LOGOS increases with feature number. However, the absolute computation time is lower for LOGOS.

was undertaken to investigate the feasibility of using the inliers determined by LOGOS to perform metric pose estimation. The KITTI dataset provides rectified stereo data, and the stereo images were used to estimate 3D position information for each point via a disparity map. The disparity map was calculated via a block matching system that compared 15×15 blocks using the sum of absolute differences (SAD).

Matches for the pose estimation step were computed by LOGOS with an empirically-determined threshold of 31 inliers; that is, images where more than 31 inliers were found were accepted. A total of 719 images fitted this criteria, out of a total of 788 possible true matches. For each of these images, the LOGOS inliers were used with the standard Levenberg-Marquardt least-squares algorithm to compute the transformation matrix between these images, and the error from the ground truth was computed, using the

error methodology from the original dataset [19].

Fig. 12 shows the error in the metric pose estimation calculated using the inlier matches from LOGOS on the KITTI dataset. The rotation error is smaller than 0.1rad in 95% of cases, and smaller than 0.2rad in 99% of cases, while the translation error is smaller than 1.8m in 95% of cases, and smaller than 3.2m in 99% in cases.



(a) Rotation (b) Position

Fig. 12. Rotation and position error in metric pose estimation using LOGOS on the KITTI dataset. In 95% of cases, the rotation error is smaller than 0.1rad and the position error is smaller than 1.8m.

For comparison, the pose estimate was also calculated using standard RANSAC with full 128-dimension descriptors. The LOGOS inliers performed comparably to the full RANSAC calculation, which had a rotation error smaller than 0.1rad in 95% of cases, and smaller than 0.2rad in 98% of cases, and a translation error smaller than 1.8m in 94% of cases, and smaller than 3.2m in 98% in cases.

The greater inaccuracy seen in the translation error for both LOGOS and RANSAC is likely due to the block matching used to calculate the 3D positions of the image points. A more sophisticated depth estimation algorithm could be implemented, such as that used in [20]; however, this simple framework demonstrates that the inlier sets calculated by LOGOS can be used to perform metric pose estimation as well as topological spatial verification. If necessary, the inlier sets calculated by LOGOS could be further refined by using a RANSAC geometric verification step to remove additional outliers.

VI. CONCLUSION

LOGOS is a method of inlier detection that can be used for spatial verification and pose estimation in visual localization systems. LOGOS performs effective location verification even when there are many outliers in the data. On benchmark visual localization datasets LOGOS consistently provided a high level of recall at 100% precision, and the performance of LOGOS was shown to degrade gracefully as the cutoff parameters increased and as the number of feature points stored decreased.

The effectiveness of LOGOS depends on the quality of the initial correspondences. The number of features stored per image is thus an important factor in LOGOS, as is the nature of the operating environment. LOGOS was evaluated on datasets largely captured in urban and suburban areas, and will also be tested in environments that contain fewer large, regular structures.

Future work for LOGOS will focus on integrating additional consistency checks with LOGOS, particularly to increase global consistency across inliers. Techniques to select useful features will also be developed, to reduce the memory requirements further without impacting performance.

REFERENCES

[1] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[2] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[3] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, June 1981.

[4] P. H. S. Torr and A. Zisserman, "MLESAC: A new robust estimator with application to estimating image geometry," *Computer Vision and Image Understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[5] O. Chum, J. Matas, and J. Kittler, "Locally optimized RANSAC," in *Pattern Recognition: 25th DAGM Symposium*, 2003, pp. 236–243.

[6] O. Chum and J. Matas, "Matching with PROSAC - progressive sample consensus," in *2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 220–226.

[7] W. Zhang and J. Kosecka, "A new inlier identification scheme for robust estimation problems," in *Proceedings of Robotics: Science and Systems*, Philadelphia, USA, August 2006.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004.

[9] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *International Journal of Computer Vision*, vol. 87, no. 3, pp. 316–336, May 2010.

[10] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza, "MAV urban localization from Google street view data," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov 2013, pp. 3979–3986.

[11] X. Li, M. Larson, and A. Hanjalic, "Pairwise geometric matching for large-scale object retrieval," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5153–5161.

[12] C. C. Chou and C. C. Wang, "2-point RANSAC for scene image matching under large viewpoint changes," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 3646–3651.

[13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, June 2008.

[14] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *2011 International Conference on Computer Vision*, Nov 2011, pp. 2548–2555.

[15] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2010.

[16] 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.

[17] ——, "Appearance-only SLAM at large scale with FAB-MAP 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[18] A. Glover, W. Maddern, M. Milford, and G. Wyeth, "FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 3507–3512.

[19] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[20] J. Engel, J. Stückler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 1935–1942.