Long-Distance Loop Closure Using General Object Landmarks

Andrew Holliday and Gregory Dudek

Abstract—Visual localization under large changes in scale is an important capability in many robotic mapping applications, such as localizing at low altitudes in maps built at high altitudes, or performing loop closure over long distances. Existing approaches, however, are robust only up to a 3x difference in scale between map and query images.

We propose a novel combination of deep-learning-based object features and hand-engineered point-features that yields improved robustness to scale change, perspective change, and image noise. We conduct experiments in simulation and in real-world outdoor scenes exhibiting up to a 7x change in scale, and compare our approach against localization using state-of-the-art SIFT features. This technique is training-free and class-agnostic, and in principle can be deployed in any environment out-of-the-box.

I. INTRODUCTION

In this work, we attempt to address the problem of performing metric localization in a known environment under extreme changes in visual scale. Our localization approach is based on the identification of objects in the environment, and their use as landmarks. By “objects” we here mean physical entities which are distinct from their surroundings and have some consistent physical properties of structure and appearance.

Many robotic applications involve repeated traversals of a known environment over time. In such applications, it is usually beneficial to first construct a map of the environment, which can then be used by a robot to navigate the environment in subsequent missions. Surveying the environment from a very high altitude allows complete geographic coverage of the environment to be obtained by shorter, and thus more efficient, paths through the environment by the surveyor. At the same time, a robot that makes use of this high-altitude map to localize may have mission parameters requiring it to operate at a much lower altitude.

One such scenario is that of performing visual surveys of benthic environments, such as coral reefs, as in [1]. A fast-moving surface vehicle may be used to rapidly map a large area of a reef. This map may then be used by a slower-moving, but more maneuverable, underwater vehicle (AUV) such as the Aqua robot [2], to navigate the reef while capturing imagery very close to the sea floor. Another relevant scenario is that of a robot performing loop closure over long distances while performing simultaneous localization and mapping (SLAM) in terrestrial environments. Loop closure, the recognition of a previously-viewed location when viewing it a second time, is key to accurate SLAM, and the overall accuracy of SLAM could be considerably improved if loop closure could be conducted across major changes in scale and perspective.

In scenarios such as the above, a robot must deal with the considerable change in visual scale between two perspectives, which in our scenarios may be 5 times or even greater. In some scenarios, such as in benthic environments, other factors may also intrude, such as colour-shifting due to the optical properties of water, and image noise due to particulate suspended in the water. Identifying scenes across such large changes in scale is very challenging for modern visual localization techniques. Even the most scale-robust techniques, such as Scale-Invariant Feature Transforms (SIFT), can only localize reliably under scale factors less than 3.

We consider that the hierarchical features computed by the intermediate layers of a convolutional neural network (CNN) [3] may prove robust to changes in scale, due to their high degree of abstraction. We propose a technique for performing metric localization across significant changes in scale by identifying and describing non-semantic objects in a way that allows them to be associated between scenes. We show that these associations can be used to localize correctly under visual scale factors of 3 and greater. The proposed system does not require any environment-specific
training, and in principle can be deployed out-of-the-box in arbitrary environments. The objects used by our system are defined functionally, in terms of their utility as scale-invariant landmarks, and are not limited to semantically-meaningful object categories. We demonstrate the approach both in simulation and on a novel dataset of real-world image pairs of urban scenes.

II. RELATED WORK

Global Localization refers to the problem of determining a robot’s location in a pre-existing map with no prior on the robot’s position [4]. This is closely related to the problem of loop closure in SLAM, and there is a large body of literature exploring this problem. Prominent early work includes Leonard et al. [5] and Mackenzie et al. [6], and [7].

Many traditional visual approaches to these problems have been based on the recognition of whole-image descriptors of particular scenes, such as GIST features [8]. Successful instances include SeqSLAM [9], which uses a heavily downsampled version of the input image as a descriptor, and FAB-MAP [10], ORB-SLAM [11], and Ho et al. [12] which discretize point-feature descriptors to build bag-of-words histograms of the input images, starting from SURF [13], ORB [14], and SIFT [15] features, respectively. Other successful results include LSD-SLAM of Engel et al. [16] as well as Hansen et al. [17], Cadena et al., [18], Liu et al. [19] and Naseer et al. [20]. Because whole-image descriptors encode the geometric relationships of features in the 2D image plane, images of the same world scene from different perspectives can have very different descriptors, in general no more similar than those of images of completely different places. Most such methods are thus very sensitive to changes in perspective, and depend on viewing the world from the same perspective in the mapping and localization stages.

Other global localization approaches attempt to recognize particular landmarks in an image, and use those to produce a metric estimate of the robot’s pose. SLAM++ of Salas-Moreno et al. [21] performs SLAM by recognizing landmarks from a database of 3D object models. Linegar et al. [22] and Li et al. [23] both train a bank support vector machines (SVMs) to detect specific landmarks in a known environment, one SVM per landmark. More recently, [24] made use of a Deformable Parts Model (DPM) [25] to detect objects for use as loop-closure landmarks in their SLAM system. All of these approaches require a pre-existing database of either object types or specific objects to operate. These databases can be costly to construct, and these systems will fail in environments in which not enough landmarks belonging to the database are present.

Some work has explored the use of CNNs for global localization. PoseNet [26] is a CNN that learns mapping from an image in an environment to the metric pose of the camera, but can only operate on the environment on which the network was trained. In Sündenrhauf et al. [27], the intermediate activations of a CNN trained for image classification were used as whole-image descriptors for place recognition, a non-metric form of global localization. Their subsequent work [28] refined this approach by using the same descriptor technique on object proposals within an image instead of the whole image. These pre-trained deep descriptors were robust to changes in appearance and perspective. Schmidt et al. [29] and Simo-Serra et al. [30] have both explored the idea of learning point-feature descriptors with a CNN, which could replace classical point features in a bag-of-words model.

When exploring robustness to perspective change, all of these works only consider positional variations of at most a few meters, when the scenes exhibit within-image scale variations of tens or hundreds of meters, and when the reference or training datasets consisted of images taken over traversals of environments ranging from hundreds to thousands of meters. As a result, little significant change in scale exists between map images and query images in these experiments. To the best of our knowledge, ours is the first to attempt to combine deep object-like features and point features into a single, unified representation of landmarks. This synthesis provides superior metric localization to either technique in isolation, particularly under significant (3x and greater) changes in scale.

III. PROPOSED SYSTEM

The first stage of our metric localization pipeline consists in detecting objects in a pair of images, computing convolutional descriptors for them, and matching these descriptors between images. Our approach here closely follows that used for image-retrieval by Sunderhauf et al. [27]; we differ in using Selective Search (SS), as proposed by Uijlings et al. [31], to propose object regions, and in our use of a more recent CNN architecture.

To extract objects from an image, Selective Search object proposals are first extracted from the image, and filtered to remove objects with bounding boxes less than 200 pixels in size and with aspect ratio greater than 3 or less than 1/3. The image regions defined by each surviving SS bounding box are then extracted from the image, rescaled to a fixed size via bilinear interpolation, and run through a CNN. We use a ResNet-50 architecture trained on the ImageNet image-classification dataset, as described in He et al. [32]. Experiments were run using six different layers of the network as feature descriptors, and with inputs to the network of four different resolutions. The network layers and resolutions are listed in Table II.

Having extracted objects and their descriptors from a pair of images, we perform brute-force matching of the objects between the images. Following [28], we take the match of each object descriptor in image \( i \) to be the descriptor in image \( j \) that has the smallest cosine distance from \( i \), as defined in [4]. Matches are validated by cross-checking; a match \((x_i, y_j)\) is only considered valid if \( x_i \) is the most similar object to \( y_j \) in image \( i \) and \( y_j \) is the most similar object to \( x_i \) in image \( j \).

\[
t_{\cos, 	ext{corr}}(u, v) = 1 - \frac{u \cdot v}{||u||_2 \cdot ||v||_2}
\]  
(1)
TABLE I: The sizes, as a number of floating-point values, of the output layers of ResNet-50 at different input resolutions. Values in bold indicate layer-resolution pairs which provided the best results in any of our experiments.

| Input res. | pool1 | res2c | res3d | res4f | res5c | pool5 |
|------------|-------|-------|-------|-------|-------|-------|
| 224 × 224  | 201k  | 803k  | 401k  | 200k  | 100k  | 2k    |
| 128 × 128  | 66k   | 262k  | 131k  | 65k   | 131k  | 8k    |
| 64 × 64    | 16k   | 66k   | 32k   | 66k   | 131k  | 8k    |
| 32 × 32    | 4k    | 16k   | 32k   | 66k   | 131k  | 8k    |

Fig. 2: A simplified illustration of our object-detection architecture.

Once object matches are found, we extract SIFT features from both images, using 3 octave layers, an initial Gaussian with \( \sigma = 1.6 \), an edge threshold of 10, and a contrast threshold of 0.04. For each pair of matched objects, we match SIFT features that lie inside the corresponding bounding boxes to one another. SIFT features are matched via their Euclidean distance, and cross-checking is again used to filter out bad matches. By limiting the space over which we search for SIFT matches to matched object regions, we greatly increase the accuracy of SIFT matching, and thus of the resulting metric pose estimates. As a baseline against which to compare our results, experiments were also run using SIFT alone, with no object proposals. In these baseline experiments, SIFT matching was performed in the same way, but the search for matches was conducted over all SIFT features in both images.

These point matches are used to produce a metric pose estimate. Depending on the experiment, we compute either a homography \( H \) or an essential matrix \( E \), or both. In either case, the calculation of \( H \) or \( E \) from point correspondences is done via a RANSAC algorithm with an inlier threshold of 6, measured in pixel units.

IV. EXPERIMENTS IN SIMULATION

A. Experimental Setup

A range of experiments were conducted in simulation that imitate the scenario of a low-altitude robot with a downward-facing camera localizing in an approximately-planar world using a visual map constructed at an altitude several times higher. These experiments were conducted in the Gazebo simulation environment, using a world consisting of a variety of publicly-available textured 3D models spread over an untextured ground plane. A simulated camera was used first to take a "map" image at a high altitude, encompassing all of the objects in the environment. Then, a set of query images were taken at a "pyramid" of poses spaced in regular 9-by-9 grids at a range of altitudes, giving rise to scale changes between the map and query image of 1.5 times to 7 times. All images taken were 1200 × 900 pixels, with a simulated focal length of 1039.47 in pixel units.

At each location, a random orientation was generated by sampling a \( \Delta \text{yaw} \), \( \Delta \text{pitch} \), and \( \Delta \text{roll} \) from a uniform distribution over the range \([-40^\circ, 40^\circ]\), and these deltas were added to the yaw, pitch, and roll of the downward-facing mapping camera’s orientation. The camera pose from which each image was taken was recorded and used as ground truth for evaluation. Images which did not contain any visible objects were filtered out of the dataset. The final simulated dataset consisted of one map image and 561 query images. Fig. 3 is the map image against which the query images were compared.

For each query image, a set of point matches was produced between the query image and the map image according to the method detailed in III, or using SIFT feature matching in the case of the SIFT-only experiments. These point matches were used to estimate both a homography matrix \( H \) and an essential matrix \( E \). Pose estimates \((R_H, t_H)\) and \((R_E, t_E)\) describing the transform between the mapping camera and the query camera, consisting of a rotation matrix and a translation vector, was derived from each of these matrices. The reprojection error of all inlier matches for each of \( H \) and \( E \) was computed from the respective pose estimates, and the pose estimate with the lower reprojection error was returned. The quality of the estimate was measured as the relative positional error, as defined in (2):

\[
t_{rel\text{err}} = \frac{||t_g - t_e||_2}{||t_g||_2}
\]

where \( t_g \) is the ground-truth translation between the two cameras and \( t_e \) is the estimated translation. We normalize the...
vector from the estimated pose to the true pose to remove the correlation of that vector’s length with the magnitude of the true translation. In degenerate cases of erroneous matches, the estimated translation can be extremely large, and in practice any $t_{rel, err}$ greater than 2 means that the magnitude of the error is twice the magnitude of the actual translation. Hence, we threshold the error above at 2:

$$\text{error} = \min(t_{rel, err}, 2) \quad (3)$$

If no pose can be estimated for a query image due to insufficient or incoherent point matches, we set the error to 2 for this query as well. To test the degree of robustness to noise of our technique, each experiment is repeated with multiple levels of Gaussian per-pixel noise applied to each image. The $\sigma$ values used are 0, 0.04, 0.08, and 0.12, where the per-pixel noise is applied to each colour channel.

B. Results

Table II displays the average error of each configuration of our system over all queries, as well as that of using SIFT features alone. Features drawn from network layer res5c with 6x6-pixel inputs gave the best performance overall, but notably, 13 of the 24 configurations - more than half - outperformed SIFT overall. Notably, for each input resolution, the deepest convolutional layers of the network - res4f, res5c, and pool5 - performed best at each input resolution. This supports the idea that more levels of abstraction from the input pixels in a descriptor leads to improved robustness to scale and perspective change.

Fig. 4 compares the performance of feature res5c at input resolution 64 × 64 to that of SIFT features alone, at each query scale factor and at each noise level. All methods have average error close to 2 at scale factors of 5 and above, implying that most queries fail to localize at these scale changes. At scale factors 2, 3, and 4, however, there is a notable difference between our system and SIFT features at all noise levels. The gap between the average error without noise and that with the highest level of noise is significantly greater for SIFT features than for our system at all scale factors less than 4, indicating that limiting the search for SIFT matches only to the regions of matching objects improves their robustness not only to changes in scale, but also to image noise.

TABLE II: The average positional error over all query images at all noise levels, for each configuration of our system, as well as for SIFT features alone. The best-performing configuration is in bold.

| Input res. | pool1 | res2c | res3d | res4f | res5c | pool5 |
|------------|-------|-------|-------|-------|-------|-------|
| 224 × 224  | 1.751 | 1.889 | 1.779 | 1.669 | 1.609 | 1.630 |
| 128 × 128  | 1.768 | 1.842 | 1.685 | 1.614 | 1.591 | 1.612 |
| 64 × 64    | 1.740 | 1.750 | 1.626 | 1.589 | 1.567 | 1.587 |
| 32 × 32    | 1.673 | 1.676 | 1.619 | 1.573 | 1.594 | 1.604 |
| SIFT       | 1.666 |

Fig. 4: The mean error vs. the scale factor relative to the map image for SIFT features and the best configuration of our system, at four levels of per-pixel noise.

V. MONTREAL IMAGE PAIR EXPERIMENTS

A. Experimental Setup

To test the effectiveness of the proposed system in a real-world scenario, a set of 31 image pairs were taken across twelve scenes surrounding the Montreal campus. Scenes were chosen to contain a roughly-centred object of approximately uniform depth in the scene, so that a uniform change in the object’s image scale could be achieved by taking images at various distances from the object. Scenes were also selected so that any objects surrounding the central object were closer to the camera than the central object, so that no part of the scene would exhibit a much smaller scale change than that of the central object. This ensures that successful matches must be made under the close to the desired change in scale, and makes the relationship between the images amenable to description by a homography. The image pairs exhibit changes in scale ranging from factors of about 1.5 to about 7, with the exception of one image pair showing scale change of about 15 in a prominent foreground object. All images were taken using the rear-facing camera of a Samsung Galaxy S3 phone, and were downsampled to 900x1200 pixels via bilinear interpolation for all experiments.

Each image pair was hand-annotated with a set of 10 point correspondences, distributed approximately evenly over the nearer image in each pair. The proposed system was used to compute point matches between each image pair, and from these point matches, a homography $H$ was computed as described in section II. $H$ was used to calculate the total symmetric transfer error (STE) for the image pair $e$ over the ground truth points:

$$\text{STE} = \sum_{i} ||p_{far}^i - Hp_{near}^i||_2 + ||p_{near}^i - H^{-1}p_{far}^i||_2 \quad (4)$$
The maximum STE we observed for any attempted method on any image pair was $STE_{\text{max}} = 15.833, 861, 380.8$. Whenever no $H$ could be found for an image pair by some method, its on that image pair was set to $STE_{\text{max}}$. The plain STE ranges over many orders of magnitude on this dataset, so we present the results using the logarithmic STE, making the results easier to interpret.

The same set of parameters were run over this dataset as in the simulation experiments - our system at six network layers and four input resolutions, plus SIFT alone for comparison.

### B. Results

Table III shows the performance of each feature layer and each input resolution over the whole Montreal dataset, and shows the results from using SIFT features alone as well. As this table shows, the total error using just SIFT features is significantly greater than that of the best-performing input resolution for each feature layer. Also, the average error of the intermediate layers res2c, res3d, and res4f, are all very comparable. It is interesting to note that in this experiment, more intermediate layers are favoured, while the simulation experiments favoured the deepest layers of the network. This may arise from the relatively small degrees of change in viewing angle between these image pairs in comparison with those in the simulation experiments.

Fig 5 show the error of each of the three best-performing configuration, as well as SIFT, on each of the image pairs in the dataset, plotted versus the median scale change over all pairs of ground-truth matches $(p^i, p^j)$ in each image. The scale change between matches $(p^i, p^j)$ is defined in [5].

The lines of best fit for each method further emphasize the improvement of our system over SIFT features at all scale factors up to 6. The best-fit lines for all of the top-three configurations of our system overlap almost perfectly, although there is a fair degree of variance in their performances on individual examples.

$$\text{scale change}_{i,j} = \frac{||p^i_{\text{near}} - p^j_{\text{near}}||_2}{||p^i_{\text{far}} - p^j_{\text{far}}||_2} \quad (5)$$

The use of homographies to relate the image pairs allows us to visually inspect the quality of the estimated $H$, by using $H$ to map all pixels in the farther image to their estimated locations in the nearer image. Visual inspection of these mappings for the 31 image pairs confirm that those configurations with lower logarithmic STEs tend to have more correct-looking mappings, although all configurations of our system with mean logarithmic STE $\leq 10$ produce comparable mappings for most pairs, and on some pairs, higher-error configurations such as res4f with $64 \times 64$-pixel inputs produce a subjectively better mapping than the lowest-error configuration. Fig. 1 and Fig. 6 display some example homography mappings.

### VI. CONCLUSIONS

We have shown that by combining deep learning with classical methods, we can recognize scenes and perform accurate localization across vast changes in scale. Our system uses a pre-trained deep network to describe arbitrary objects and correctly match them between images for use as navigation landmarks. Restricting SIFT feature matching to matched objects substantially improves the robustness of SIFT features both to changes in image noise and to changes in scale. Despite much prior work on place recognition and localization using both classical methods and deep learning, our result sets a new benchmark for metric localization performance across scale.

One strength of our proposed system is that it requires no domain-specific training, making use only of a pre-trained CNN. However, as future work we wish to explore the possibility of training a CNN with the specific objective of producing a perspective-invariant object descriptor, as doing so may result in more accurate matching of objects. We also wish to explore the possibility that including matches from multiple layers of the network in the localization process could improve the system’s accuracy.

### TABLE III: A table showing the logarithmic STE of each configuration of the system, averaged over all image pairs.

| Input res. | pool1 | res2c | res3d | res4f | res5c | pool5 |
|------------|------|-------|-------|-------|-------|-------|
| $224 \times 224$ | 11.102 | 13.057 | 9.631 | 10.277 | 10.537 | 10.011 |
| $128 \times 128$ | 10.389 | 11.716 | 10.231 | 9.458 | 9.930 | 9.505 |
| $64 \times 64$ | 10.921 | 9.381 | 9.339 | 9.777 | 10.234 | 9.667 |
| $32 \times 32$ | 10.134 | 10.162 | 10.473 | 9.658 | 10.301 | 10.607 |
| SIFT | 10.654 |

Fig. 5: The error of SIFT features alone, and the three best-performing configurations of our system, on each image pair in the dataset, plotted versus the median scale change exhibited in the image pair, along with a line of best fit for each method.
Fig. 6: The results of SIFT features and three different configurations of our system on another pair. The homography estimated by the best-overall configuration, res3d at 64 \times 64 configurations of our system on another pair. The homography input size, is notably worse than those produced by two other intermediate feature layers. No configuration of our system performs best on all image pairs.