GPS-Denied UAV Localization using Pre-existing Satellite Imagery

Hunter Goforth and Simon Lucey[1]

***Abstract*— We present a method for localization of Unmanned Aerial Vehicles (UAVs) which is meant to replace an onboard GPS system in the event of a noisy or unreliable GPS signal. Our method requires only a downward-facing monocular RGB camera on the UAV, and pre-existing satellite imagery of the flight location to which the UAV imagery is compared and aligned. To overcome differences in the image capturing conditions between the satellite and UAV, such as seasonal and perspective changes, we propose the use of Convolutional Neural Network (CNN) representations trained on readily available satellite data. To increase localization accuracy, we also develop an optimization which jointly minimizes the error between adjacent UAV frames as well as the satellite map. We demonstrate how our method improves on recent systems from literature by achieving greater performance in flight environments with very few landmarks. For a GPS-denied flight at 0.2km altitude, over a flight distance of 0.85km, we achieve average localization error of less than 8 meters. We make our source code and datasets available to encourage further work on this emerging topic[2].**

I. INTRODUCTION

The commercial and consumer use of UAVs outdoors has grown exponentially in the last few years. UAVs now find use in search and rescue [1], [2], industrial inspection [3], [4], land surveying and mapping [5], [6], precision agriculture [7], monitoring of remote environments [8], [9], the study of wildlife populations [10], and many more. Nearly all applications require precise latitude and longitude estimates of the UAV during flight, with some also requiring accurate altitude or 6 degrees-of-freedom (DoF). This level of localization may not be available in GPS-denied or GPS-spoofed situations, and requires the use of often costly GPS hardware and IMUs. A vision-based system that achieves comparable accuracy would be beneficial as it would lower the cost of UAV platforms, and could replace GPS when there are signal issues. This idea is further motivated by the fact that there is ample free, GPS-aligned, satellite imagery covering many parts of the globe available online. This satellite imagery can provide a map prior for a flight, against which we can perform template matching to localize.

The main challenge of using satellite imagery as a map prior is overcoming the differences in imaging conditions between the satellite images and the incoming video stream from the UAV. An example of this is shown in Fig. 1. Since the UAV images are captured at much lower altitude than the satellite map, there is a larger perspective effect of structures higher than the ground plane. The correspondence of typical feature descriptors like SIFT [11], which are commonly used in the remote sensing community for satellite-to-satellite image matching [12], [13], [14], usually fails with such perspective differences. Another challenge can come from temporal aspects, such as seasonal effects, time-of-day, and the removal or addition of buildings or cars. Overcoming these temporal aspects has been attempted using learned, sparse descriptors in [15]. However, sparse descriptors are still reliant on local texture and inherently cannot generalize to more rural, low texture flight environments.

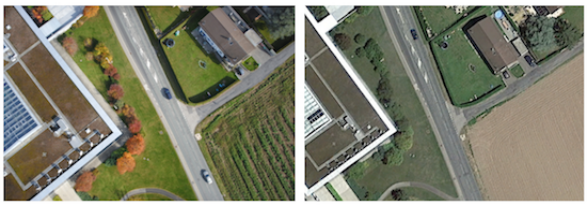

Fig. 1. A typical example of alignment between UAV imagery (left) and a satellite image (right). Notice differences such as seasonal variation of vegetation, shadow angles, perspectives of buildings, presence of vehicles, and variations due to different imaging hardware. Our goal is overcoming these variations to enable precise UAV localization.

Prior Work. There have been a few attempts in recent literature to develop a full UAV localization system based on satellite image matching. In [16], the authors use HOG features for alignment between UAV imagery and the map, making their approach reliant on well-defined, temporally consistent texture such as buildings and roadways. In [17], the authors accomplish alignment using a mutual information metric. They show the method working only on a UAV flight over a textured urban environment, and the satellite map they use is photometrically very similar to the UAV imagery captured. In [18], the authors develop a complex pipeline for alignment which includes SIFT matching and semantic segmentation of buildings. This makes their method heavily reliant upon the presence of texture, as well as on the photometric similarity of UAV images and the map. They must also train their neural network semantic segmentation on data extracted from the exact location and with similar imaging qualities as used at test time. To give an idea of the performance of these systems, each achieves average localization error of less than approximately 10 meters across flight distances of between 300-1500 meters, and at altitudes of 100-300 meters. All previous work makes the flat-world assumption in order to parameterize motion using the planar homography, a model that we will also use.

Contributions. We show through experimentation that our method can achieve comparable or better performance than prior work, with three important improvements. First, our method can generalize from urban environments to

[1]Authors are with The Robotics Institute at Carnegie Mellon University, {hgoforth,slucey}@andrew.cmu.edu

[2]https://github.com/hmgoforth/gps-denied-uav-localization

challenging low-texture rural datasets. Second, though our method is based on neural networks, we train only on freely available satellite imagery, and not on any data from the UAV used at test time. Third, we introduce an optimization over the pose of the UAV at all frames, which incorporates constraints from visual odometry as well as the satellite map alignment, allowing accurate localization even at UAV frames which are not directly compared with the map.

Overview. In general, our method involves three steps. First, we perform visual odometry to determine initial estimates of motion parameters. Next, we compare a subset of recent frames to the satellite map to geolocalize those frames (described in Section II). Finally, we refine the geolocalized pose of all frames using a joint optimization between frame odometry and map alignment (described in Section III). In Section IV, we provide localization results in an urban environment, and in a rural location with few landmarks.

II. ALIGNING SATELLITE IMAGES

For the task of aligning UAV images with satellite images, we choose to employ a recently introduced method in which a deep CNN representation is learned for direct alignment [19], [20]. We hypothesize that features can be learned which are effective for aligning satellite imagery and UAV images under many different imaging conditions such as seasonal changes, time-of-day, perspective differences, and the addition or removal of buildings or other structures. We believe direct alignment using all image pixels is an effective choice as we can utilize the global texture of an image when performing alignment, which has been shown important for aligning low texture imagery [21]. Most importantly, we show that using this method allows us to learn features only from satellite imagery that is freely available online, and not from imagery from a particular UAV or location used at test time.

A. Direct Alignment with Learned Features

The network architecture in both [19] and [20] consists of two fully-convolutional neural networks in parallel with the same convolutional weights, through which the images to be aligned are passed. After the convolutional network, the feature images are passed into a differentiable implementation of the Inverse Compositional Lucas-Kanade (ICLK) algorithm [22]. Loss functions are designed in order to learn the weights of the convolutional layers, such that the feature images passed to the ICLK layer are as photometrically similar as possible. [20] expands on this architecture with a coarse-to-fine approach, learning separate features at each coarseness level. In [19], the authors fine-tune AlexNet convolutional filters, while in [20] they have enough data to train a network from scratch.

For our implementation, we have chosen to use the output features of the 3rd convolutional block of the VGG16 network [23], and fine-tune the weights only in the 3rd block. We choose the 3rd block based on the known generalizibility characteristics of mid-layer features in large image recongition neural networks [24]. We show that only a limited amount of satellite data is needed to effectively learn features for alignment. We implement the ICLK layer using full projective alignment with homography, with the geometric corner loss function described in [20].

B. Training

To fine-tune features suitable for aligning satellite imagery and UAV imagery, we predict that only a small, representative batch of satellite imagery is necessary. Therefore our training data is gathered from the imagery available on the United States Geographical Survey Earth Explorer website[3]. We gather data from a geographical area of 5.9km by 7.5km, in New Jersey, USA. Some representative images from this dataset are shown in Fig. 2. We chose the location for the fact that it has an even mix of urban, suburban, and rural imagery, capturing low and high texture. The imagery is gathered during the years 2006, 2008, 2010, 2013, 2015, and 2017, across Spring, Summer and Fall. In total there are 10 large images, each 7582×5946 pixels at a resolution of 1 meter per pixel.

During training, we dynamically create training pairs by extracting random patches from images in the New Jersey Dataset. We create a training batch by randomly choosing 2 of the 10 large images, and then choosing a random latitude, longitude, altitude, and compass heading for extracting a patch from both images. Keeping one of the patches static, we warp the other patch using random projective warp coordinates. We then perform the forward pass in the network using the created training batch, and update the convolutional weights by backpropagating from the value of the corner loss across the training batch. Further details can be found in Section V.

C. Testing

We reserve 20% of the geographical area from the New Jersey Dataset for testing. This allows us to test the generalization of the features learned, for an unseen area of the map. The result of this test is shown in Fig. 3. This test is indicative of the generalization performance of our learned features, for aligning satellite imagery. We make the assumption that learning features that are robust between temporally varying satellite images, is a similar task to learning features that are robust between a satellite image and imagery from a high-altitude UAV. However, the testing of this hypothesis comes in the localization experiments in Section IV. We find impressive generalization to UAV imagery, despite the relatively small amount of training data.

III. POSE PARAMETER OPTIMIZATION

With the method of aligning UAV imagery to satellite imagery shown in Section II, we have one part of the whole system. For some applications, this alignment capability may be enough on its own. In the previous works on these systems [16], [17], [18], finding a suitable method of alignment is the main focus, be it mutual information, SIFT, HOG, semantic shape matching, etc.

[3]https://earthexplorer.usgs.gov/

Fig. 2. Some examples from the New Jersey alignment training dataset. Source image patches (top) and their corresponding template images (bottom) are taken from separate large orthographic satellite images covering an area of 5.9km by 7.5km. The dataset contains many images from both urban locations and low-texture rural locations.

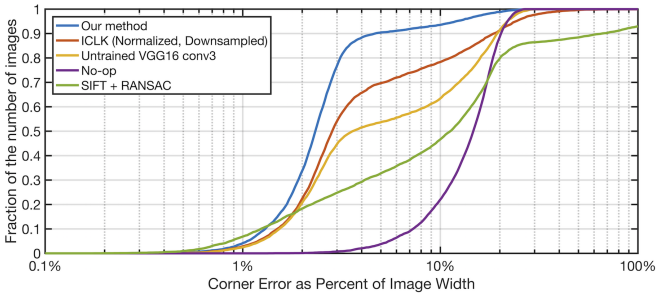

Fig. 3. Testing learned features for the task of ICLK image alignment. We compare our method (blue) with standard ICLK on zero-mean, unit variance normalized image pairs (red), using vanilla, untrained VGG16 conv3 features (yellow), and an implementation of SIFT+RANSAC alignment (green). No-op (purple) is the corner error if no attempt at alignment is made. We report the corner error as a percent of image width; having less than 3-4% in this metric results in misalignment that is usually visually imperceptible. We see that for this dataset, we are able to learn features with less than 4% error for nearly 90% of image pairs. This is most notably compared to a representative sparse, feature-based approach such as SIFT+RANSAC, which is unable to handle the differences in imaging conditions and achieves less than 4% error on only 30% of the testing dataset (for SIFT implementation details, refer to Section V).

However, in the Simultaneous Localization and Mapping (SLAM) literature, a common concept is optimizing over all the pose parameters throughout an image sequence. Techniques such as bundle adjustment or pose graph optimization attempt to optimize the pose of the camera and/or landmarks as the camera traverses an environment, as in [25], [26], [27], [28]. Specifically, we draw inspiration from the recent introduction of photometric bundle adjustment, which optimizes directly on pixel intensity [29], [30]. In our case, we optimize on the Sum of Squared Differences (SSD) between pixel intensities of temporally adjacent UAV frames, as well as the SSD between (deep feature extractions of) UAV frames and the satellite map. The primary benefit of this approach is that it allows us to precisely localize the UAV at all frames, using only a smaller subset of frames to match to the satellite map.

A. Formulation

Our goal is to obtain accurate, absolute pose of the UAV at any frame F during the flight sequence. We make the flat world assumption and parameterize the motion of the UAV in terms of planar homographies with respect to the flat world. Assuming we have a satellite map that is aligned to GPS coordinates, then the absolute pose of the UAV at any frame can be encoded in a homography relating the satellite map image to the current frame, which we call H^F_{abs}. Then, the goal is to estimate H^F_{abs} for all frames.

We assume the position of the UAV is approximately known at the time of capturing the initial frame $F = 1$. This is a reasonable assumption in applications where the approximate take-off position is known, or where a single GPS data point is given, before beginning GPS-denied flight. There is an absolute homography H^1_{abs} relating the initial view of the UAV to the satellite map image at frame $F = 1$, whose parameters can be determined based on the approximate heading and GPS location of the UAV. As the UAV moves, each frame F is related to the last frame $F-1$ by a relative homography H^F_{rel} computed using image pairs from the UAV (visual odometry). Therefore, the absolute homography relating the view from the UAV at frame F to the satellite map is the composition of the intial absolute homography H^1_{abs} with all relative homographies up to frame F:

$$H^F_{abs} = H^F_{rel} \cdot H^{F-1}_{rel} \cdot ... \cdot H^2_{rel} \cdot H^1_{abs} \tag{1}$$

We define certain frames throughout the UAV sequence as template frames T, to which other temporally adjacent frames are aligned. We define visibility neighborhoods V around each template, which contain the frames adjacent to T. V should be chosen so that there is sufficient overlap between all frames in V with T so as to allow direct registration such as with Lucas-Kanade. The precise choice of template frames and visibility neighborhoods depends on many characteristics of the particular flight hardware and the speed of the vehicle, and must be tuned for a given application. A graphical depiction of the optimization variables is shown in Fig. 4.

To align with the map, we extract the deep feature representation of all T using our trained convolutional layers from Section II. We also extract the patch from the map M which corresponds to T based on the current estimate of all motion parameters in the sequence, and extract the deep features of this map patch. Then, we simultaneously optimize for the motion parameters based on the minimization of error between all templates T and the images within their corresponding visibility neighborhoods V, as well as the error between the deep feature extractions of the templates T and the map M. We express this minimization objective as:

$$\begin{aligned}\min_{\mathbf{p}} \sum_k \sum_{F \in V(k)} ||I_F(W(\mathbf{x}; \mathbf{p}_{F,k})) - T_k(\mathbf{x})||_2^2 + \\ \lambda \sum_k ||M^\phi(W(\mathbf{x}; \mathbf{p}_{M,k})) - T_k^\phi(\mathbf{x})||_2^2\end{aligned} \tag{2}$$

Where k is the set of frames which are templates, I_F is the UAV image at frame F, M^ϕ and T^ϕ are deep feature extractions of the map and template respectively, $\mathbf{p}_{F,k}$ is the composition of motion parameters from frame F to template

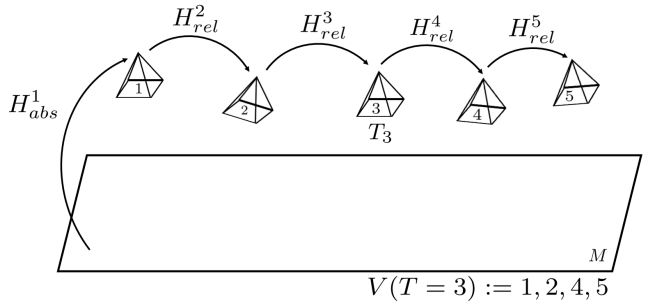

Fig. 4. Visualization of the parameters for optimization. H^1_{abs} parameterizes the approximately known initial pose of the UAV, at the time that GPS-denied flight begins. Initial estimates of H^F_{rel} are obtained via visual odometry. This example uses a single template T_3 and single corresponding visibility neighborhood V, but many templates with overlapping neighborhoods can be used in practice. The template is the only UAV frame which is compared against the satellite map M.

k, and $\mathbf{p}_{M,k}$ is the composition of motion parameters from the satellite map M to the template k. We use the notation $I(W(\mathbf{x};\mathbf{p}))$ to mean sampling an image I at image coordinates $\mathbf{x}$ that have been warped with warping function W, using a projective transformation parameterized by $\mathbf{p} \in \mathbb{R}^8$. λ is a tunable parameter used to weight the contribution of the map alignment; if it is zero, the optimization does not use the satellite map at all, but optimizes the motion parameters based only on UAV imagery (visual odometry).

B. Derivation of Motion Parameter Update

The minimization in (2) can be solved using the iterative Gauss-Newton method. We start by defining $\mathbf{p}_f$ of a particular relative homography in the UAV sequence H^f_{rel} that we optimize with respect to. We rewrite both $\mathbf{p}_{M,k}$ and $\mathbf{p}_{F,k}$ as functions of $\mathbf{p}_f$ with the addition of a small perturbation $\Delta\mathbf{p}_f$ to the motion parameters:

$$\sum_k \sum_{F \in V(k)} ||I_F(W(\mathbf{x};\mathbf{p}_{F,k}(\mathbf{p}_f + \Delta\mathbf{p}_f))) - T_k(\mathbf{x})||_2^2 + \lambda \sum_k ||M^\phi(W(\mathbf{x};\mathbf{p}_{M,k}(\mathbf{p}_f + \Delta\mathbf{p}_f))) - T_k^\phi(\mathbf{x})||_2^2 \quad (3)$$

Linearizing with respect to $\mathbf{p}_f$ yields

$$\sum_k \sum_{F \in V(k)} ||J_I \Delta\mathbf{p}_f + r_I||_2^2 + \lambda \sum_k ||J_M \Delta\mathbf{p}_f + r_M||_2^2 \quad (4)$$

where

$$J_I = \nabla I_F \frac{\partial W}{\partial \mathbf{p}_{F,k}} \frac{\partial \mathbf{p}_{F,k}}{\partial \mathbf{p}_f}, \quad J_M = \nabla M^\phi \frac{\partial W}{\partial \mathbf{p}_{M,k}} \frac{\partial \mathbf{p}_{M,k}}{\partial \mathbf{p}_f} \quad (5)$$

$$r_I = I_F(W(\mathbf{x};\mathbf{p}_{F,k}(\mathbf{p}_f))) - T_k(\mathbf{x}) \quad (6)$$

$$r_M = M^\phi(W(\mathbf{x};\mathbf{p}_{M,k}(\mathbf{p}_f))) - T_k^\phi(\mathbf{x}) \quad (7)$$

We note that $\frac{\partial \mathbf{p}_{F,k}}{\partial \mathbf{p}_f}$ is nonzero only if $f \in V(k)$ and $|f-k| \leq |F-k|$, and $\frac{\partial \mathbf{p}_{M,k}}{\partial \mathbf{p}_f}$ is nonzero only if $f \leq k$. Finally, taking the derivative w.r.t. $\Delta\mathbf{p}_f$, setting the minimization equal to zero, and solving for $\Delta\mathbf{p}_f$ yields:

$$\Delta\mathbf{p}_f = -\mathcal{H}^{-1}\left(\sum_k \sum_{F \in V(k)} J_I^T r_I + \lambda \sum_k J_M^T r_M\right) \quad (8)$$

where

$$\mathcal{H} = \sum_k \sum_{F \in V(k)} J_I^T J_I + \lambda \sum_k J_M^T J_M \quad (9)$$

The optimal update $\Delta\mathbf{p}_f$ for all relative homographies H^f_{rel} can be computed in parallel using this iterative solution, with the update at each iteration $\mathbf{p}_f \leftarrow \mathbf{p}_f + \Delta\mathbf{p}_f$. The iteration continues until the maximum value of $\Delta\mathbf{p}_f$ across all frames is below a certain threshold.

The optimization can be applied in a sliding-window or a batch fashion. A sliding-window approach would be more suitable for applications with a requirement for online localization. We present experiments using a sliding-window, with more details in the following section.

IV. LOCALIZATION EXPERIMENTS

As few attempts have been made at UAV localization systems similar to the one presented here, there are no freely available datasets or baseline implementations. A proper dataset for this task consists of a UAV sequence using a monocular, downward-facing camera, with precise ground truth pose with respect to the Earth, and an accompanying globally aligned satellite image of the flight location. The altitude of the UAV must also be such that the flat-world assumption can reasonably be made. Of the few previous works [16], [17], [18], none have made their datasets or source code publicly available for comparison. Therefore we gather two datasets, in part using UAV imagery from the free datasets offered from the senseFly professional drone mapping company[4]. The first dataset is captured overhead the Swiss village of Merlischachen, at an altitude of 0.2km, over 0.85km flight distance. This is an urban environment similar to those used in all prior works, with ample texture and landmarks such as buildings and roadways. The second dataset is more challenging, captured overhead a rural gravel quarry at 0.22km altitude and flight distance of 0.61km. This location offers substantially less texture and landmarks for alignment. We use this dataset to showcase the capabilities of our system to align with low-texture imagery, an advantage over past approaches.

A. Village Dataset

Some example UAV frames from the Village dataset and an overview of the flight are in Fig. 5. The UAV used is the senseFly eBee drone, equipped with a downward facing Canon IXUS 125 HS camera. We extract the GPS-aligned satellite map from Google Earth Pro. The UAV imagery was captured in April, 2013, and the satellite imagery in May, 2012. High-accuracy RTK GPS (latitude, longitude, and altitude) is included in the metadata of each UAV frame in the dataset. We process the GPS metadata of the initial

[4]https://www.sensefly.com/education/datasets/

frame of the dataset in order to form H^1_{abs}, the homography relating the initial UAV pose to the satellite map, as described in Section III-A. To form initial estimates of the frame-to-frame relative motion H^F_{rel}, we use SURF [31] feature extraction and correspondence to estimate homographies between UAV frames. We find SURF provides odometry performance comparable to other popular sparse-feature based methods, and is thus used as a representative baseline.

With estimates for H^1_{abs} and H^F_{rel}, we solve for the optimal motion parameters of the UAV using (8). For the Village dataset, we find empirically that using a sliding-window approach for the pose optimization produces the lowest average localization error. This is due to the low frame-rate of the UAV dataset, where there is as low as 50% frame-to-frame overlap. This makes the composition of H^F_{rel} from the SURF odometry inaccurate when predicting pose after many frames. It is this predicted position which is used to extract patches from the map M in the optimization to use for alignment, but if these patches do not have significant overlap with the UAV frames then the optimization fails. We use a sliding-window containing 4 UAV frames, of which the 2nd and 4th frame images are the templates T, with $V(2) = 1, 3$ and $V(4) = 3$ as the visibility neighborhoods of the two templates. This window is slid forward by two frames after the optimal parameters are found. We start with the first 4 frames of the video, and progress through the whole sequence. With this sliding window, only half of all video frames are directly compared with the map.

After solving for the optimal motion parameters, we convert them to GPS coordinates (latitude, longitude, and altitude) and plot error from ground truth, versus SURF odometry. Localization results on the Village dataset are shown in Fig. 6, and some example alignments in Fig. 7. Despite relatively little training data as described in Section II, the learned features generalize well to the unseen UAV imaging conditions and satellite map used in both the Village and Gravel Pit datasets. We find other alignment methods simply do not come close to proper alignment on these datasets, including ICLK on normalized images, or ICLK on vanilla conv3 VGG16 features. SIFT produces very few (3 to 5), if any correct correspondences between UAV images and satellite images, not enough for effective RANSAC estimation.

B. Gravel Pit Dataset

Example frames from the Gravel Pit dataset are shown in Fig. 8, along with a flight overview. The same UAV and camera are used from the Village dataset. The UAV imagery was captured in April, 2013, and the satellite map we extract is from November, 2014. We chose this dataset to showcase the abilities of our method of alignment for use on very low-texture environments lacking landmarks. The effects of seasonal variation are more pronounced in this dataset due to increased vegetation as well.

We repeat the process from the Village dataset for extracting initial motion parameter estimates. For the motion parameter optimization, we experiment by eliminating the

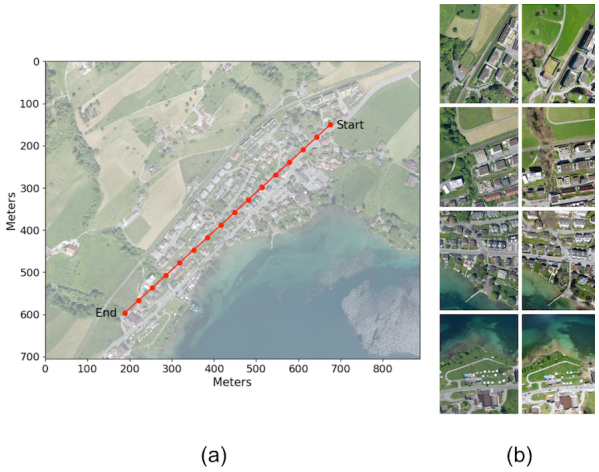

Fig. 5. (a) Overview of the Village dataset flight path. (b) Some examples of UAV frames (right) and their corresponding satellite map patches (left) for the Village dataset.

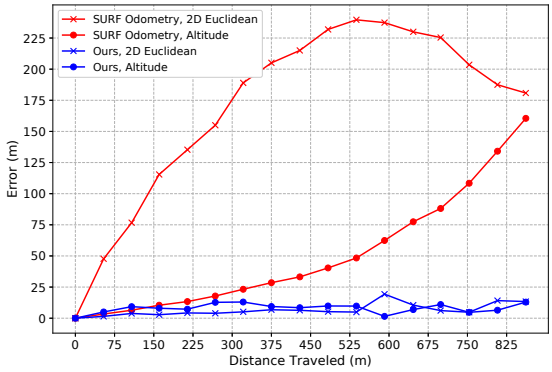

Fig. 6. Our results on the Village dataset. Results labeled 2D Euclidean are error distances measured in the x-y plane. Markers represent template frames. The average 2D euclidean error of our method is 7.06m. The average altitude error of our method is 8.01m. The ground truth altitude of the UAV for the entire sequence is approximately 0.2km. We find that alignment methods including SIFT+RANSAC or ICLK on vanilla VGG16 conv3 features fail to align UAV images with the satellite map for this dataset, and thus do not improve on the error from SURF odometry.

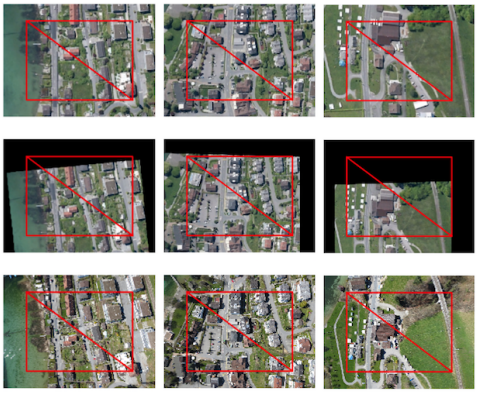

Fig. 7. Some examples of the alignment capabilities of our system on the Village dataset, which occur during optimization of motion parameters. Crops from the satellite map (top) are warped and aligned (middle) with the corresponding UAV imagery (bottom). The red rectangle is static in all images, allowing the reader to visually compare aligned landmarks.

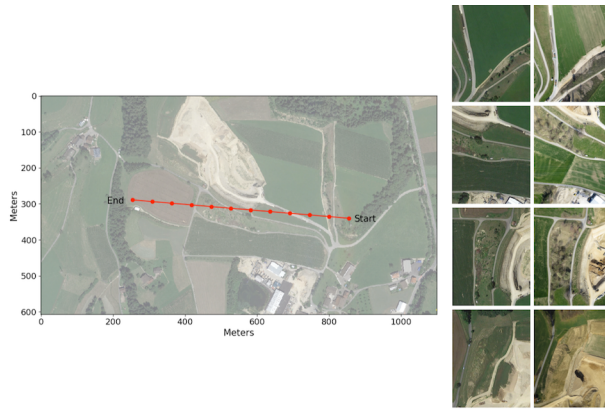

Fig. 8. (a) Overview of the Gravel Pit dataset flight path. (b) Some examples of UAV frames (right) and their corresponding satellite map patches (left) for the Gravel Pit dataset.

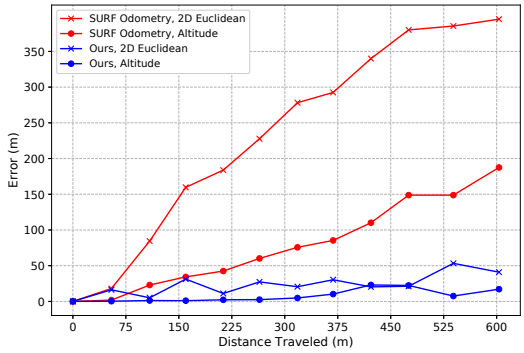

Fig. 9. Our results on the more challenging, low texture Gravel Pit dataset. The average 2D euclidean error of our method is 25.00m. The average altitude error of our method is 7.70m. Precise x-y localization is difficult on this dataset due to low texture, leading to ambiguous alignment. However, no other methods we try for UAV-map matching are able improve on the odometry error as ours has.

inclusion of the odometry term in (2) and using only the minimization of the map with templates. Further, we use a sliding-window of size 1 (a single template), and move the sliding window by 1 frame each step. This way, the optimization will seek to fully align all UAV frames with the map. This is equivalent to what is done in prior work, where visual odometry is not taken into consideration for computing the optimal motion parameters. We use this approach to illustrate that the algorithm recovers from misalignment on one frame and successfully relocalizes in the next frame, even with large UAV motion between frames. Results are shown in Fig. 9, with some example alignments in Fig. 10.

V. Implementation Details

Training and testing in Section II uses Python 3.6 and PyTorch 0.3.0. We train using square satellite image patches sized between 175-225 pixels. Randomly generated projective warp parameters for training include in-plane rotations (simulated yaw) of up to 20 degrees, image scaling between

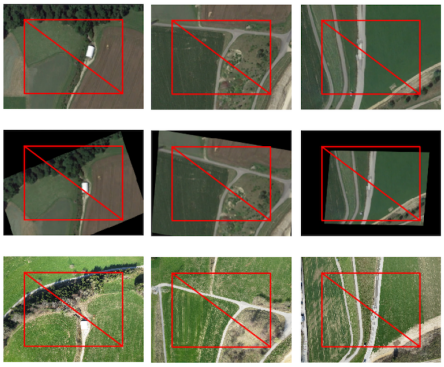

Fig. 10. Some examples of our alignment on the Gravel Pit dataset. Crops from the satellite map (top) are warped and aligned (middle) with the corresponding UAV imagery (bottom). The red rectangle is static in all images, allowing the reader to visually compare aligned landmarks.

0.75 and 1.25, translation of up to 20 pixels, and out-of plane (simulated roll and pitch) rotation of up to 5 degrees. Training is done over 50,000 mini-batches of 10 image pairs each. We use the Adam optimizer with learning rate $1e^{-4}$. We use an OpenCV implementation of SIFT and SURF detection and computation, with brute-force matcher and cross-check for smallest distance match. We also use the OpenCV's `findHomography` function with RANSAC, with an error threshold of 5 pixels. The satellite map for the Village dataset is 4800×2861 pixels at 0.45 meters per pixel width. The map for the Gravel Pit dataset is 3355×1852 at 0.32 meters per pixel width. Native UAV image resolution is 4608×3456. UAV images and map patches are square-cropped and scaled to 200 pixels each during pose optimization. The output of VGG16 conv3 has 256 feature channels and is 3-times spatially downsampled from the input. On a 2.9 GHz Intel Core i5 laptop with 16 GB RAM, optimization using 2 templates with 4 frames in each visibility neighborhood, and two overlapping frames between neighborhoods, takes 8.41s on average over 100 trials. Computational efficiency can be improved for practical implementations by using inverse-composition, C++ with optimization libraries, and with GPU. We empirically find $\lambda = 0.35$ best for optimization experiments in these particular datasets.

VI. Conclusion

We present a method for GPS-denied localization of UAVs, making several contributions that improve on previous work, noted at the end of Section I. In an effort to encourage more methods and baselines in this space, we make our source code and datasets available for download. Future work can include further exploration of learning photometric invariance using the described alignment method. Instead of fine-tuning the convolutional weights of an object recognition model, a massive dataset could be compiled consisting of millions of satellite image examples and a model could be trained from scratch. Also to be explored is a rigorous method for the selection of templates T and visibility neighborhoods V for the optimization, based on characteristics of the UAV flight that could be automatically determined.

REFERENCES

[1] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An autonomous multi-uav system for search and rescue," in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, ser. DroNet '15. New York, NY, USA: ACM, 2015, pp. 33–38. [Online]. Available: http://doi.acm.org/10.1145/2750675.2750683

[2] J. Qi, D. Song, H. Shang, N. Wang, C. Hua, C. Wu, X. Qi, and J. Han, "Search and rescue rotary-wing uav and its application to the lushan ms 7.0 earthquake," *Journal of Field Robotics*, vol. 33, no. 3, pp. 290–321, 2016.

[3] S. Omari, P. Gohl, M. Burri, M. Achtelik, and R. Siegwart, "Visual industrial inspection using aerial robots," in *Applied Robotics for the Power Industry (CARPI), 2014 3rd International Conference on*. IEEE, 2014, pp. 1–5.

[4] J. Cacace, A. Finzi, V. Lippiello, G. Loianno, and D. Sanzone, "Aerial service vehicles for industrial inspection: task decomposition and plan execution," *Applied Intelligence*, vol. 42, no. 1, pp. 49–62, 2015.

[5] S. Siebert and J. Teizer, "Mobile 3d mapping for surveying earthwork projects using an unmanned aerial vehicle (uav) system," *Automation in Construction*, vol. 41, pp. 1–14, 2014.

[6] J.-n. Okamoto and H. Shimazaki, "Land surveying performance of commercially-available inexpensive small uav," in *Proceedings of 36th Asian Conference on Remote Sensing 2015 (ACRS 2015), Fostering Resilient Growth in Asia*. Citeseer, 2015.

[7] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic uav and ugv system for precision agriculture," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1498–1511, 2016.

[8] Y. Lu, D. Macias, Z. S. Dean, N. R. Kreger, and P. K. Wong, "A uav-mounted whole cell biosensor system for environmental monitoring applications," *IEEE Trans. Nanobiosci*, vol. 14, pp. 811–817, 2015.

[9] M. Messinger and M. Silman, "Unmanned aerial vehicles for the assessment and monitoring of environmental contamination: An example from coal ash spills," *Environmental pollution*, vol. 218, pp. 889–894, 2016.

[10] J. C. Hodgson, S. M. Baylis, R. Mott, A. Herrod, and R. H. Clarke, "Precision wildlife monitoring using unmanned aerial vehicles," *Scientific reports*, vol. 6, p. 22574, 2016.

[11] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[12] M. Gong, S. Zhao, L. Jiao, D. Tian, and S. Wang, "A novel coarse-to-fine scheme for automatic image registration based on sift and mutual information," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 7, pp. 4328–4338, July 2014.

[13] Y. Bentoutou, N. Taleb, K. Kpalma, and J. Ronsin, "An automatic image registration for applications in remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 9, pp. 2127–2137, Sept 2005.

[14] K. Yang, A. Pan, Y. Yang, S. Zhang, S. H. Ong, and H. Tang, "Remote sensing image registration using multiple image features," *Remote Sensing*, vol. 9, no. 6, 2017. [Online]. Available: http://www.mdpi.com/2072-4292/9/6/581

[15] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "Tilde: A temporally invariant learned detector," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[16] M. Shan, F. Wang, F. Lin, Z. Gao, Y. Z. Tang, and B. M. Chen, "Google map aided visual navigation for uavs in gps-denied environment," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2015, pp. 114–119.

[17] A. Yol, B. Delabarre, A. Dame, J. Dartois, and E. Marchand, "Vision-based absolute localization for unmanned aerial vehicles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sept 2014, pp. 3429–3434.

[18] A. Nassar, K. Amer, R. ElHakim, and M. ElHelw, "A deep cnn-based framework for enhanced aerial imagery registration with applications to uav geolocalization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 1513–1523.

[19] C. Wang, H. K. Galoogahi, C. Lin, and S. Lucey, "Deep-lk for efficient adaptive object tracking," in *2018 IEEE Conference on Robotics and Automation (ICRA)*, 2017. [Online]. Available: http://arxiv.org/abs/1705.06839

[20] C. H. Chang, C. N. Chou, and E. Y. Chang, "Clkn: Cascaded lucas-kanade networks for image alignment," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 3777–3785.

[21] H. Alismail, B. Browning, and S. Lucey, "Robust tracking in low light and sudden illumination changes," in *2016 Fourth International Conference on 3D Vision (3DV)*, Oct 2016, pp. 389–398.

[22] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework," *International journal of computer vision*, vol. 56, no. 3, pp. 221–255, 2004.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2014, pp. 3320–3328. [Online]. Available: http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf

[25] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment - a modern synthesis," in *International workshop on vision algorithms*. Springer, 1999, pp. 298–372.

[26] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, "A fast and accurate approximation for planar pose graph optimization," *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 965–987, 2014.

[27] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 3607–3613.

[28] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[29] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[30] H. Alismail, B. Browning, and S. Lucey, "Photometric bundle adjustment for vision-based slam," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 324–341.

[31] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.