A Deep CNN-Based Framework For Enhanced Aerial Imagery Registration with Applications to UAV Geolocalization

Ahmed Nassar[1,2], Karim Amer[2], Reda ElHakim[2], Mohamed ElHelw[2]
[1]IRISA institute, Université Bretagne Sud
[2]Center for Informatics Science, Nile University
ahmed-samy-mohamed.nassar@irisa.fr
{k.amer, r.mostafa, melhelw} @nu.edu.eg

Abstract

In this paper we present a novel framework for geolocalizing Unmanned Aerial Vehicles (UAVs) using only their onboard camera. The framework exploits the abundance of satellite imagery, along with established computer vision and deep learning methods, to locate the UAV in a satellite imagery map. It utilizes the contextual information extracted from the scene to attain increased geolocalization accuracy and enable navigation without the use of a Global Positioning System (GPS), which is advantageous in GPS-denied environments and provides additional enhancement to existing GPS-based systems. The framework inputs two images at a time, one captured using a UAV-mounted down-looking camera, and the other synthetically generated from the satellite map based on the UAV location within the map. Local features are extracted and used to register both images, a process that is performed recurrently to relate UAV motion to its actual map position, hence performing preliminary localization. A semantic shape matching algorithm is subsequently applied to extract and match meaningful shape information from both images, and use this information to improve localization accuracy. The framework is evaluated on two different datasets representing different geographical regions. Obtained results demonstrate the viability of proposed method and that the utilization of visual information can offer a promising approach for unconstrained UAV navigation and enable the aerial platform to be self-aware of its surroundings thus opening up new application domains or enhancing existing ones.

1. Introduction

The proliferation of Unmanned Aerial Vehicles (UAVs), also known as aerial drones, has been shifting from military applications to utilization in domestic markets. This conception came through recent developments and accessibility to robust embedded hardware platforms, miniaturized electronics and sensor modules including accelerometers, barometers, and gyroscopes, as well as introduction of high-performance processors with low power consumption and efficient batteries. Currently, UAV use is ubiquitous with applications in photography, aerial mapping, agriculture, surveillance, search and rescue, parcel delivery, to name a few. Most of these aerial platforms comprise an onboard camera and use GPS and route planning software to plan and manage navigation.

Figure 1: A figure showing the registration of two images from different domains. Left: an image from a UAV, Right: a satellite image.

Presently, the abundance Earth Observation (EO) high-resolution imagery acquired using aerial or satellite sources and covering most of the globe has facilitated the emergence of new applications. In autonomous vision-only UAV navigation, UAV camera feed is compared with aerial/satellite imagery to consequently infer drone location. However, processing these images is computationally demanding especially if data labeling is needed to extract actionable information. Furthermore, the ability to correlate two images of the same location but acquired from different sources is challenging but central for accurate navigation. In this typical image registration problem, shown in Figure 1, the challenges that arise when dealing with EO-UAV image registration can be attributed to: (1) different camera position, orientation and illumination conditions during the image acquisition phase resulting in different object appearance as well as occlusion/exposure problems that confuse traditional feature-based image registration using local features such as SIFT [34]/SURF [6]/etc., (2) dissimilarity in camera intrinsic parameters introduces photogrammetric differences between the image pair, and (3) difference in image acquisition history may result in mismatch between the image pair due to objects appearing/disappearing nence making registration more difficult. Consequently, there has been a myriad of research work related to geolocalization [36, 61, 60, 38, 58], navigation [51, 10, 62], and change detection [29, 16, 63].

Conventionally, UAV navigation has relied on its on-board sensors, such as inertial sensors and gyroscopes, and more recently on Global Navigation Satellite Systems (GNSSs), such as GPS and GLONASS, to acquire position information. However, by the time it reaches earth, and due to the large distance traveled, the satellite transmitted L1 signal power is limited -160 dBW when measured by receiver units, which is below the ambient background noise of many places on earth especially in urban locations in and around certain areas of large cities. In addition to background noise, L1 signals are also vulnerable to radio interference, GPS spoofing [57, 27, 64], and loss of Line of Sight (LOS). GPS-denied environments hence refer to areas where GPS signal is not available, jammed or too weak to be used reliably. GPS denial necessitates finding alternative reliable approaches for UAV navigation [28, 9, 4, 35, 39].

This paper presents a vision-based framework that geolocalizes a UAV using only its on-board camera to allow the aerial platform to navigate autonomously without relying on GPS signal availability. The proposed framework thus enables navigation in GPS-denied areas and can also provide enhancements to existing GPS modules. In addition, on-board visual information processing provides the UAV with real-time contextual information of its surroundings which could be beneficial in new application domains and endeavors. The main contribution of this work can be summarized as a novel framework combining traditional computer vision techniques with deep learning networks for performing satellite (reference) image and UAV (target) image registration for enhanced UAV localization. To this end, a preliminary UAV localization phase is applied followed by a novel Semantic Shape Matching (SSM) phase for UAV localization. The former is based on local hand-crafted features, i.e. SIFT (Section 3.2) for system calibration and ORB [49] for path calculation (Section 3.3). The latter phase applies semantic segmentation (Section 3.4) to both reference and target images to extract and match meaningful shapes, such as buildings and roads, understand the context, and subsequently use this information to significantly improve UAV localization accuracy (Section 3.5). It should be noted that the proposed framework relies on online mapping services such as Google Satellite, Bing Maps and Open Street Map (OSM) for reference images. In the Experiments and Results section (Section 4), the proposed work is tested on two datasets: an existing dataset and an extended one where it has been shown that, in both cases, the framework successfully geolocalizes two different UAV video feeds in different geographical locations.

2. Related Work

2.1. Vision-Based UAV Geolocalization

Using traditional computer vision techniques, image features are computed and used to compare reference and target images based on some similarity measures as presented by [13, 54]. [10, 62] use template matching with cross correlation where the aerial image is used as a template to match against another georeferenced image. These methods work only on images that are nearly identical and dont cater for significant image differences making them unusable in practical localization applications especially in dense urban areas or long journeys.

In Simultaneous Localization and Mapping (SLAM) [12], UAV navigation is achieved by acquiring and correlating distinct features of indoor or outdoor regions to map the UAV to a specific location. This method is more suitable to constrained and pre-mapped environments and has been successful mainly with micro-UAVs and quadcopters. Other work uses local feature detectors, such as SIFT [34] or SURF [6] to produce features that are subsequently matched, providing an affine transformation between reference and target images as investigated in [26, 52, 53]. Pose estimation using deep learning is used in [43], however unlike our work, their algorithm requires training with data acquired from locations similar to the area of navigation.

Significant work has also been proposed on using measurements from the UAV Inertial Measurement Unit (IMU) and on-board camera to acquire hybrid features and store them in a feature database to be used later for localization [36, 9, 46]. During flight, the feature database is queried and extracted features are used to compute UAV location using feature matching. Viswanathan et al. [60] use street-level imagery for localization. In this case, panoramic ground images are warped to top-down view images which are then compared to the UAV image using local feature matching. [31, 1] employ Convolutional Neural Networks (CNNs) to extract features for geolocalization. In Sebastien et al. [29], the comparison of multiple different image sources is achieved by applying a Siamese Network [8]. To this end, a Siamese Network is trained with two types of images, satellite and perspective-transformed panoramic

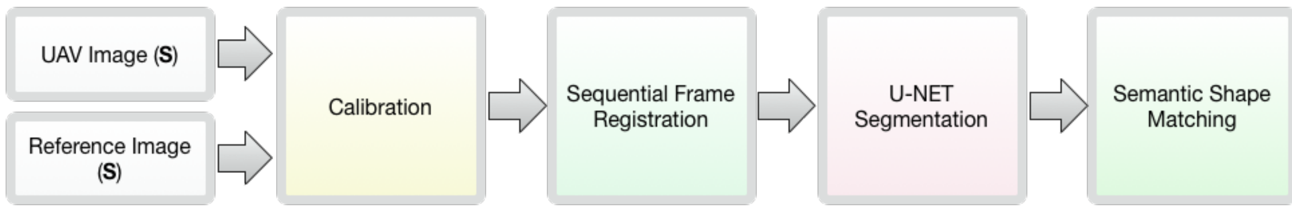


Figure 2: The major components of the proposed UAV geolocalization framework.

images. Subsequently, features from both images are extracted and the distance between the features of each image calculated to find out if they are similar. [32] finds a sequence of UAV images inside a reference map to create image mosaics. Inspired by this work, we employ this method to find transformations to provide us with the geolocation of the UAV and its path.

It is worth mentioning that the nature of the problem we are trying to solve limits the the application of recent CNN-based approaches for pose estimation [24, 44] and image-to-image regressors [50, 59]. For instance, there aren't many publicly-available aerial videos, existing ones cover limited geographical locations, and the aerial platforms usually fly at high altitudes making extraction of meaningful depth information challenging. The same applies for retrieval-based and structure from motion methods [17, 7] which require large features database and efficient retrieval methods. Our method can be deployed to new locations by training the system using available satellite imagery only and is not based solely on image retrieval or search methods.

2.2. Segmentation for Earth Observation Imagery

Earth Observation (EO) applications include weather forecast, disaster risk management, water management, coastal erosion assessment, land cover classification, maritime monitoring, natural resource management, agricultural monitoring, among others. Recently, deep learning has been applied for EO image registration and geolocalization. Semantic segmentation of EO imagery using fully convolutional deep networks has been used for EO image analysis. These algorithms seek to classify each pixel within the EO image to common classes such as surfaces (pavements, hard surfaces), buildings, vegetation, trees, cars, roads, etc. [2, 40, 48, 42]. In proposed work, the Semantic Shape Matching algorithm described in (Section 3.5) relies on semantic segmentation for detecting objects of different types.

3. Proposed Enhanced Geolocalization Framework

The proposed work relies solely on on-board camera for accurate UAV geolocalization. It comprises a sequence of key components as illustrated in Figure 2. The following sections present details of each of the framework components.

3.1. Input Images

The framework processes image sequences from two different sources: real images acquired from the UAV (aerial images) and synthetic images generated from the satellite imagery (reference map images). For the latter, multiple satellite imagery sources are used to be able to provide adequate visual description of the covered geographical area.

Aerial Images (UAV) This work focuses on vision-based geolocalization and excludes UAV control commands. The framework accepts a video sequence from the UAV denoted as $\boldsymbol{S}$. From $\boldsymbol{S}$, we can extract $\boldsymbol{S_{(i)}}$ (video frame) which we compare to a reference map ($\boldsymbol{R}$). It is important to note that the initial starting GPS coordinate of the UAV is assumed to be known and can be defined as the center pixel of $\boldsymbol{S_{(1)}}$. This assumption is made based on notion that a UAV cannot be deployed without knowing its location (at least initially).

Reference Map Images (Satellite) The framework uses a reference map $\boldsymbol{R}$ with known GPS bounds. Mainly the process is finding out where $\boldsymbol{S_{(i)}}$ resides in $\boldsymbol{R}$ and subsequently estimating the position of $\boldsymbol{S_{(i)}}$. Using Equations 1a & 1b, it is possible to calculate a certain GPS coordinate (lat, lon), by knowing $\boldsymbol{R}$s width, height, and the extent or bounds of $\boldsymbol{R}$ such as $\boldsymbol{lon_n}$, $\boldsymbol{lat_e}$, $\boldsymbol{lat_w}$, and $\boldsymbol{lon_s}$. The opposite is also possible to estimate the latitude and longitude of a pixel ($\boldsymbol{pix_w}$,$\boldsymbol{pix_h}$) using Equations 1c & 1d.

$$pix_x = \frac{(width_{max} - width_{min})(lon - lon_w)}{(lat_s - lat_w)} \tag{1a}$$

$$pix_y = \frac{(height_{max} - height_{min})(lat - lat_n)}{(lon_s - lon_n)} \tag{1b}$$

$$lat = \frac{lat_n + (lat_n - lat_s)(pix_h - height_{min})}{(height_{max} - height_{min})} \tag{1c}$$

$$lon = \frac{lon_w + (lon_e - lon_w)(pix_w - width_{min})}{(width_{max} - width_{min})} \tag{1d}$$

3.2. Calibration

The Calibration component is responsible for computing an affine geometric transformation between the UAV image and the reference map image in order to map the UAV onto a reference map location, i.e. map $\boldsymbol{S}_{(i)}$ with its field of view extents to corresponding position and covered region in $\boldsymbol{R}$, taking into consideration the difference in scale and orientation between the two images. Calibration is an integral part of the framework, and is called upon on at $\boldsymbol{S}_{(1)}$ and every three iterations ($\boldsymbol{S}_{(i+3)}$) frames to autocorrect any drift that might occur. To accomplish this task, SIFT feature points are extracted from the UAV image and the reference image and subsequently matched. Even though SIFT feature computation is time consuming, the calibration process is applied every several frames to maintain real-time performance. As previously stated, the initial GPS coordinates of the UAV are assumed to be known and so a region of interest $\boldsymbol{r}$ can be cropped from the reference map $\boldsymbol{R}$ where $\boldsymbol{r} = \boldsymbol{R}(\boldsymbol{l}, \boldsymbol{w}_{\boldsymbol{S}+\boldsymbol{b}})$ is the coordinate of the center pixel. The region width $\boldsymbol{w}_{\boldsymbol{S}+\boldsymbol{b}}$ is equal to the width of $\boldsymbol{S}$ plus a margin $\boldsymbol{b}$. This is done to limit the search of the features extracted from $\boldsymbol{S}_{(i)}$ in $\boldsymbol{R}$ to include only features in sub region $\boldsymbol{r}$ thus reducing the search space and eliminating the possibility of matching to go astray. It is important to note that $\boldsymbol{r}$ covers a wider region than the one in $\boldsymbol{S}$ and that Equations 1a, 1b, 1c, and 1d provide the mapping from pixel locations to map coordinates and vice versa.

Using the created reduced search space, features are extracted from both $\boldsymbol{r}$ and $\boldsymbol{S}_{(i)}$. Subsequently, the features are matched and a statistical estimator, such as RANSAC [14], is used to estimate a homography matrix that describes the transformation between UAV and reference image pairs. During UAV flight, the Calibration procedure is applied every several frames to sustain reasonably accurate translation, rotation and scale components between UAV images and reference map images.

3.3. Sequential Frame Registration

An image registration process is applied on sequential UAV images, $\boldsymbol{S}_{(i)}$ and $\boldsymbol{S}_{(i+f)}$ with $\boldsymbol{f}$ being the processing frame rate, for UAV egomotion estimation and subsequently mapping this motion to the reference map. The processing frame rate is used to adapt registration rate to platform translation while sustaining real-time performance. Furthermore, since consecutive frames usually don't entail significant changes between the images, efficiently computed ORB [49] features are used for key point extraction and matching. Figure 4 shows ORB features extracted from two successive UAV frames and used in the registration process in a way similar to the method used in the Calibration component. It is important to note that even though ORB is an efficient alternative to SURF and SIFT, it cant handle scale variance robustly. Therefore, scale estimation and subsequent UAV altitude estimation, depend mainly on the Calibration phase. As shown in Figure 4, after matching the key ORB points a homography is estimated which provides us with the translation in pixels. This translation then updates $\boldsymbol{l}$ using 1a & 1b.

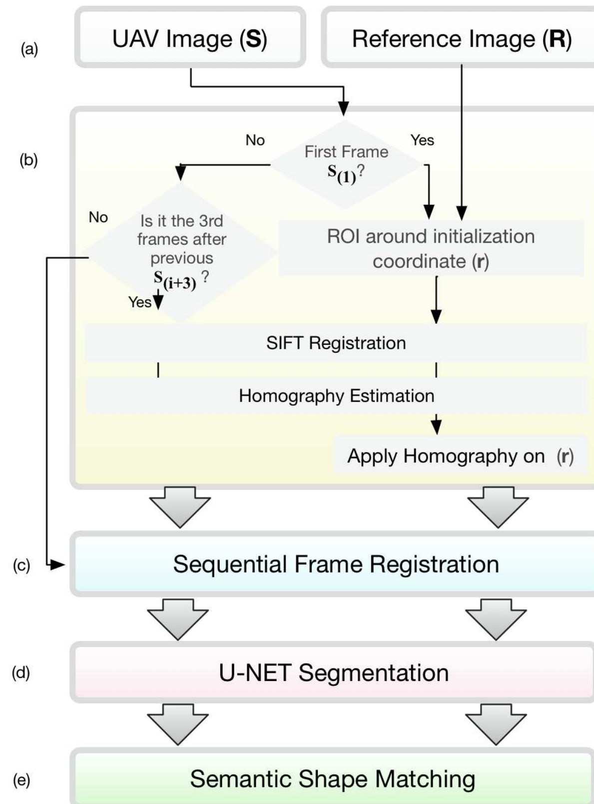


Figure 3: If the condition of being the first frame, or the 3rd sequence frame, $\boldsymbol{S}$ and $\boldsymbol{r}$ are passed to the SIFT registration, which gives out a homography, in which it is applied to $\boldsymbol{r}$ and updates the UAV location as well.

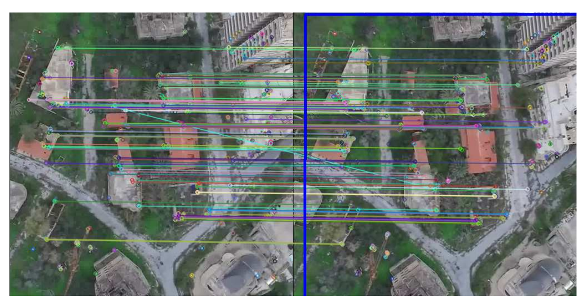


Figure 4: Registration of $\boldsymbol{S}_{(i)}$ and $\boldsymbol{S}_{(i+1)}$ using matched ORB features.

3.4. Semantic Segmentation Using U-Net

Semantic segmentation is utilized to help extract meaningful shapes in UAV and satellite images, such as roads and buildings, and pass them to the **SSM** component for matching. To be able to accomplish this, $S_{(i)}$ and r are fed into U-Net [47] network. U-Net is an encoding and decoding model that was originally proposed for biomedical image segmentation. It is chosen due its performance and prominence in recent EO tasks [30] in comparison to FCN [33] and SegNet [5]. In fact adding skip connections to FCN [23] provides finer segmentation which is already implemented in U-Net. This enables the classification of our image pixels into many classes. However, for registration, we rely on building and road classes as shown in Figure 5.

Implementation Details The original U-Net network is slightly modified where regularization using dropout layers is added after every convolutional layer with a value of 0.5. After every dropout layer, a batch normalization layer is also added to normalize the activation at each batch, which resulted in small improvement ($\approx 3\%$) in segmentation results. We also experimented with freezing the first convolutional layer as will be explained in Section 4.

Figure 5: Buildings segmented using U-Net (Left: an EO image. Right: EO image overlaid with segmentation results).

Training The resolution of images intended to train the network exceeded 2000x2000 pixels per image. A Region of Interest (ROI) of 500x500 is thus cropped around the center pixel. Afterwards, the images are split into smaller patches of size 224x224 pixels, similar to the size used by VGG [55] and ResNet [19]. Simple normalization is applied on each patch by dividing each pixel value by its channels highest value. After prediction, the patches are stitched to form the full image. The same process is applied to ground truth images. In this work, a pre-trained model is used for initialization since pre-trained models, even trained on different datasets, provide useful initialization due to the low-level features learned at early network layers such as edges and blobs [23]. The used pre-trained model was trained using a random image generator that also creates the ground truth for the randomly generated images.

As for the the optimization algorithm, we chose the Adaptive Moment Estimator (ADAM) [25]. In our early experiments, ADAM converged much faster than stochastic gradient descent and NADAM [11]. The loss function used is Dices coefficient, also known as Srensen index [56], which computes the similarity of objects in image segmentation by finding the overlap between ground-truth objects and the output provided by the segmentation method.

$$\text{Dice score} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{2}$$

As shown in Equation 2, the Dice score is calculated by finding the true positives and penalizing the false positives as well as the true positives the method could not find. Dice is similar to the Jaccard index [21], another commonly used loss function for image segmentation, however the former is more intuitive than the latter which only counts true positives once in both the numerator and denominator. The average training epochs were 7 with a learning rate of 0.0001. Training was stopped when the average F-score of the validation step stopped improving.

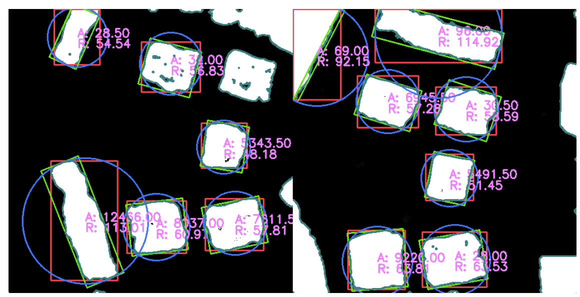


Figure 6: Left: segmented $S_{(i)}$. Right: segmented r. The contours found in a segmented image, along with the area, and radius calculated.

3.5. Semantic Shape Matching (SSM)

Applying the semantic segmentation step to $S_{(i)}$ and r results in two images containing only buildings and roads as seen in Figure 6. Next, the two images are each split into two layers: building and road objects. Morphological techniques and contouring are applied to each layer to find the boundaries of different objects and extract shape descriptor information to be used in the matching process such as location, area, roundedness, etc. In fact, knowledge of object types provides information on the surroundings of the UAV which could be used recurrently to improve segmentation results and can also be beneficial in some applications such surveillance, parcel delivery, etc. The **Semantic Shape Matching SSM** pipeline accepts a classified image with a box drawn around each contour and executes the following process (all steps are applied similarly to both images):

1. Morphological operations such as dilation followed by erosion are used to fill gaps in blobs or shapes [18].
2. Small shapes or blobs are filtered using the area of the shape found using area filtering, and using erosion followed by dilation which is another morphological operation that removes outlier pixels or noise.
3. A dictionary is built containing all the different shape features extracted. These features are the shapes' area, location, radius, centroid, and orientation, which are all calculated using Hu moments [20].
4. Using brute force matching, a scoring system is implemented to pick the matching pair. Each matching pair is awarded points based on if they have similar features within a certain tolerance. These features are also weighted due to their importance, for example matching shapes' area is more important than their centroid.
5. Matches found are checked to see if a shape matched more than once, then these matches are filtered once more based on the scoring system devised by giving higher points to certain combinations of similar features.

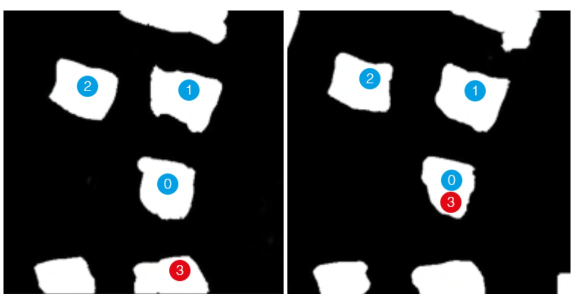


Figure 7: Matches of segmented buildings. Left: $\boldsymbol{S}_{(i)}$ and on the right $\boldsymbol{r}$. Blue circles represent positive matches, and red circles represent false positives.

The scoring system gives higher points to certain combinations of similar features. For instance, more points are awarded to matched shapes if they share the same area, distance, and orientation, rather than sharing the same area, centroid, and radius. Afterwards, matches with the highest scores are selected. However, matches are eliminated if the score is near similar and they are within a short distance since this might be an unreliable match between two objects that are of the same size and within the same distance, as shown in Figure 7. Finally, matched shapes are used to calculate a homography that is then applied to adjust the current location of the UAV with improved accuracy.

4. Experiments and Results

The framework is evaluated over two different cities for which the datasets are created manually. The semantic segmentation component is first evaluated separately since the Semantic Shape Matching **SSM** component relies on the quality achieved from the semantic segmentation step. The full framework including the **SSM** component is evaluated afterwards.

4.1. Datasets

Two cities were chosen based on the availability of the data, Famagusta (Cyprus) and Potsdam (Germany). Two dataset types are created for each city, one for training the semantic segmentation network and another for the geolocalization experiment. While creating the datasets, it was important to make sure that differences between the two image sources (UAV and satellite imagery) are minimized by taking into consideration: (1) acquisition dates are close to avoid introducing/removing different objects, and (2) using near similar image acquisition/generation parameters.

Geolocalization Dataset Due to lack of an established dataset to benchmark UAV navigation in open areas using on-board camera, a dataset had to be created. We are aware of other datasets such as Zurich Urban Micro Aerial Vehicle Dataset [37], but unfortunately a dataset does not exist with top-down view and a new dataset had to be created along with its ground-truth. For Potsdam, a simulated aerial flight video with a bird-eye view using Google Earths photorealistic 3D flight simulator was used to capture a length of 2 minutes with traveled distance of 1.2 km at an altitude of 300m. For Famagusta, a YouTube video of a UAV flight for a distance of 0.5 km was utilized. For every second in each video, a GPS coordinate was computed to create ground truth path to evaluate the proposed framework predictions.

Training Dataset To train a network capable of semantically segmenting two different sources, such as S and R, a huge dataset had to be compounded. Firstly, we used Potsdam ISPRS semantic labeling benchmark dataset[1] as our base which covers an area of 2.16 km^2. Then, we proceeded to create similar RGB tiles with the same bounds as the ISPRS dataset using Google Satellite Imagery covering the same area of 2.16 km^2. Afterwards, we extended the areas outside the tiles in ISPRS by downloading an area of 1.24 km^2 around Potsdam and 1.13 km^2 around Famagusta from Google Satellite and Bing Maps. Similar to [3, 23], the extended dataset ground-truth was created by using OpenStreetMap to train and test the semantic segmentation network. Furthermore, 30% of each UAV video sequence that are not used during the geolocalization dataset from each

[1] http://www2.isprs.org/commissions/comm3/wg4/ semantic-labeling.html

Method	Average	Building	Roads
HUSTW3 †	95.25 %	96.7 %	93.8 %
Experiment **I**	**84.7 %**	85.1 %	**84.3 %**
Kaiser et Al. [23]	83.85 %	**91.3 %**	76.4 %

Table 1: Comparison to other work using ISPRS Potsdam 2D Semantic Labeling Challenge (RGB images only). The first score is highlighted in bold. † This entry is added for a full picture as it is the highest score available on the ISPRS Potsdam 2D Challenge, however this method uses the DSM data in addition to RGB.

city was also labeled using OpenStreetMap when suitable or manually and then added to the dataset. It is important to note that these images were also augmented horizontally and vertically along with random rotations.

4.2. Semantic Segmentation Experiments

In this section, the setup of the experiment will be presented with the choices and reasons made for our network. Due to hardware availability limitations, each region (Famagusta, Potsdam) had its separate model and each class (buildings, roads, etc.) trained separately using the modified U-Net. To evaluate the semantic segmentation component of the framework, the following experiments had been carried out. It is worth mentioning that the average F1-score is used as an error metric for each trained class.

Experiment I. Was trained purely on ISPRS Potsdam dataset using 19 images for training and 5 for validation. The purpose of this experiment was to find out how well U-Net performs in comparison to other benchmark networks. The experiment was only run for the buildings and roads classes which is the most important for this region since it is mainly urban area. In general, Experiment I model performed better across the 2 classes which resulted in an average score of 84.7% as seen in Table 1. Unfortunately, the buildings class score was lagging but this experiment was the basis on which the other experiments are built upon. **Experiment II.** After testing the model generated from Experiment I on the UAV sequence, the results were not satisfactory. So, Experiment II was trained on the extended dataset (Section 4.1) using Experiment I model as the pre-trained model. The first convolutional layer was frozen while training since the first layer contains the basic shapes and edges. The scores were satisfactory when tested and the predictions provided sharp edges with hollow shapes. **Experiment III.** Carried out to validate if freezing Experiment II first convolutional layer would improve results. Therefore, in this experiment unlike Experiment II the first convolutional layer was unfrozen. Experiment III model provided fuller shapes but with inaccurate edges in comparison to Experiment II. However, this model provided the highest scores. To demonstrate the procedure of training with the online map services using OSM as ground truth, **Experiment IV** was arranged. As expected, the results were behind Experiment II, and III by nearly 10% which demonstrates that high-resolution data with accurate pixel-level ground truth definitely had a positive effect. In general, the proposed framework aim is not to introduce a new segmentation method, but to utilize the most accurate algorithm in the segmentation component of the framework. Although Experiment III provided the highest score, practically, Experiment II was the model used due to its crisp shapes while artifacts were remedied using morphological operations. Qualitatively, the buildings that are close to each other are challenging to segment separately. There is also segmentation discrepancy between buildings and sidewalks which are sometimes considered part of the buildings class. Another point to consider when using OSM to create ground-truth, is that OSM labels treats trees and vegetation as one class.

Method	Average	Building	Roads
Experiment **II**	<u>86.59 %</u>	<u>87.98 %</u>	**85.2**
Experiment **III**	**85.7 %**	**88 %**	<u>83.4 %</u>
Experiment **IV**	*75.5 %*	*76.3 %*	*74.7%*

Table 2: The first score is highlighted in bold, the second in underline, and third in italic.

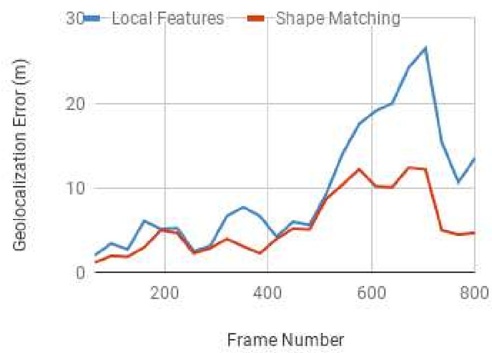
Figure 8: Ground truth and estimated paths. Top: Potsdam. Bottom: Famagusta. Left: Local features. Right: SSM.

4.3. Geolocalization Experiments

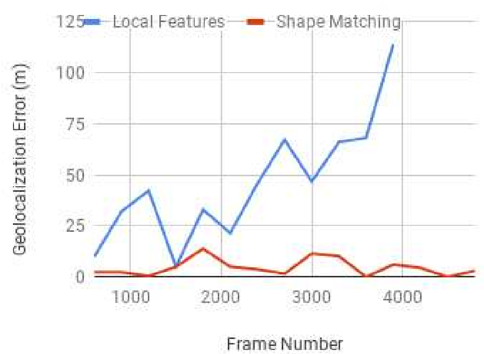
The proposed integrated framework is evaluated for UAV geolocalization using $\boldsymbol{S}$ and $\boldsymbol{r}$. Since there are no similar work the authors are aware of, the framework performance is evaluated by breaking it down to geolocalization (1) using local features (Calibration and Sequential Registration components), and (2) using the full pipeline (including the SSM component). The distance between the predicted GPS coordinates and the ground truth is the main evaluation metric chosen to represent drift or geolocalization error.

Using Local Features The performance of this approach depends on the quality of local features extracted from $\boldsymbol{S}$ and $\boldsymbol{r}$, and used for geolocalization, as explained in Section 3. It was found that this method achieves an average drift of 10.4m and 6.3m from the ground truth in the Famagusta and Potsdam datasets, respectively. Qualitatively, as seen in Figures 8a & 8c, using only local features, the deviation or geolocalization error is apparent.

Using SSM This method takes $\boldsymbol{S}_{(i)}$ and its corresponding $\boldsymbol{r}$, and matches the equivalent segmented shapes to improve registration. The path estimated by the framework is compared to ground-truth of manually-labelled GPS coordinates. The geolocalization error from the ground-truth is reduced to a 5.1m error for Famagusta, and considerably less for Potsdam to 3.6m as shown in Figures 8b & 8d. This result could be explained by the low quality of Famagustas $\boldsymbol{R}$. As for Potsdam, the quality of both $\boldsymbol{R}$, and $\boldsymbol{S}$ contributed to better localization as shown in the results in Figures 9a & 9b. This clearly demonstrates that applying scene contextual information extracted by semantic segmentation and semantic shape matching significantly reduce geolocalization error. However, it has been observed that in Potsdam, building blocks are tightly spaced so when segmented, they create huge blobs that are very difficult to match at low altitudes. Better localization could be achieved in less dense areas, at higher altitudes, or with finer segmentation.



(a) Famagusta



(b) Potsdam

Figure 9: These figures show the Geolocalization Error in distance from the ground truth for the local features and shape matching techniques.

5. Conclusion

This paper presented a novel framework for geolocalizing a UAV using its on-board camera and EO imagery. The vision-only approach enables the UAV to navigate without a GPS hence alleviating GPS spoofing and denial problems. The components of the proposed framework are explained including a novel matching method **SSM** that performs registration by matching semantically segmented objects thus improving upon methods using local features. This semantic segmentation model was compared against similar work and also evaluated as part of the framework.

Carried out experiments demonstrate that (1) incorporating online mapping information as additional data sources for training semantic segmentation networks increases the accuracy of segmentation and (2) using shape and contextual information provide improved geolocalization than relying solely on local features.

One possible future work direction is to experiment with larger datasets covering multiple cities and to improve the semantic segmentation pipeline by exploiting additional UAV videos. This dataset should prove beneficial when developing an end-to-end deep learning algorithm that replaces the traditional computer vision methods used in this work. It is also expected that adding more classes to semantic segmentation outputs will increase the robustness of the framework. Will also investigate efficient semantic segmentation methods such as ESPNet [41], [45], and ShuffleSeg [15]. As for the **SSM** component, it can be further improved by replacing the demanding dictionary search and heuristics by a CNN inspired by [22]. Pose estimation and image-to-image registration can also be experimented with for Calibration and sequential frame registration components.

Acknowledgments: This work was done while the first author was at Nile University.

References

[1] K. Amer, M. Samy, R. ElHakim, M. Shaker, and M. ElHelw. Convolutional neural network-based deep urban signatures with application to drone localization. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 2138–2145. IEEE, 2017.

[2] N. Audebert, B. Le Saux, and S. Lefvre. Semantic Segmentation of Earth Observation Data Using Multimodal and Multi-scale Deep Networks. In *Computer Vision ACCV 2016*, pages 180–196. Springer, Cham, Nov. 2016.

[3] N. Audebert, B. L. Saux, and S. Lefèvre. Joint learning from earth observation and openstreetmap data to get faster better semantic maps. *arXiv preprint arXiv:1705.06057*, 2017.

[4] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy. Estimation, planning, and mapping for autonomous flight using an rgb-d camera in gps-denied environments. *The International Journal of Robotics Research*, 31(11):1320–1343, 2012.

[5] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[6] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[7] A. Bergamo, S. N. Sinha, and L. Torresani. Leveraging structure from motion to learn discriminative codebooks for scalable landmark classification. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 763–770. IEEE, 2013.

[8] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah. Signature verification using a" siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, pages 737–744, 1994.

[9] G. Chowdhary, E. N. Johnson, D. Magree, A. Wu, and A. Shein. Gps-denied indoor and outdoor monocular vision aided navigation and control of unmanned aircraft. *Journal of Field Robotics*, 30(3):415–438, 2013.

[10] G. Conte and P. Doherty. Vision-based unmanned aerial vehicle navigation using geo-referenced information. *EURASIP Journal on Advances in Signal Processing*, 2009:10, 2009.

[11] T. Dozat. Incorporating nesterov momentum into adam. 2016.

[12] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[13] B. Fan, Y. Du, L. Zhu, and Y. Tang. The registration of uav down-looking aerial images to satellite images with image entropy and edges. *Intelligent Robotics and Applications*, pages 609–617, 2010.

[14] 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*, 24(6):381–395, 1981.

[15] M. Gamal, M. Siam, and M. Abdel-Razek. Shuffleseg: Real-time semantic segmentation network. *arXiv preprint arXiv:1803.03816*, 2018.

[16] N. Ghouaiel and S. Lefèvre. Coupling ground-level panoramas and aerial imagery for change detection. *Geo-spatial Information Science*, 19(3):222–232, 2016.

[17] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu. 3d visual phrases for landmark recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3594–3601. IEEE, 2012.

[18] R. M. Haralick, S. R. Sternberg, and X. Zhuang. Image analysis using mathematical morphology. *IEEE transactions on pattern analysis and machine intelligence*, (4):532–550, 1987.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[20] M.-K. Hu. Visual pattern recognition by moment invariants. *IRE transactions on information theory*, 8(2):179–187, 1962.

[21] P. Jaccard. *Le coefficient generique et le coefficient de communaute dans la flore marocaine*. Impr. Commerciale, 1926.

[22] L. Jiang, T. Zhao, C. Bai, A. Yong, and M. Wu. A direct fingerprint minutiae extraction approach based on convolutional neural networks. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 571–578. IEEE, 2016.

[23] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 2017.

[24] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2938–2946. IEEE, 2015.

[25] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] T. Koch, P. d'Angelo, F. Kurz, F. Fraundorfer, P. Reinartz, and M. Korner. The tum-dlr multimodal earth observation evaluation benchmark. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2016.

[27] L. Kugler. Why gps spoofing is a threat to companies, countries. *Communications of the ACM*, 60(9):18–19, 2017.

[28] S. Lange, N. Sunderhauf, and P. Protzel. A vision based onboard approach for landing and position control of an autonomous multirotor uav in gps-denied environments. In *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pages 1–6. IEEE, 2009.

[29] S. Lefèvre, D. Tuia, J. D. Wegner, T. Produit, and A. S. Nassar. Toward seamless multiview scene analysis from satellite to street level. *Proceedings of the IEEE*, 2017.

[30] R. Li, W. Liu, L. Yang, S. Sun, W. Hu, F. Zhang, and W. Li. Deepunet: A deep fully convolutional network for pixel-level sea-land segmentation. *arXiv preprint arXiv:1709.00201*, 2017.

[31] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5007–5015, 2015.

[32] Y. Lin and G. Medioni. Map-enhanced uav image sequence registration and synchronization of multiple image sequences. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007.

[33] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[34] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[35] Y. LY, H. Zhang, et al. A robust real-time vision based gps-denied navigation system of uav. 2016.

[36] A. L. Majdik, Y. Albers-Schoenberg, and D. Scaramuzza. Mav urban localization from google street view data. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3979–3986. IEEE, 2013.

[37] A. L. Majdik, C. Till, and D. Scaramuzza. The zurich urban micro aerial vehicle dataset. *The International Journal of Robotics Research*, 36(3):269–273, 2017.

[38] A. L. Majdik, D. Verda, Y. Albers-Schoenberg, and D. Scaramuzza. Air-ground matching: Appearance-based gps-denied urban localization of micro aerial vehicles. *Journal of Field Robotics*, 32(7):1015–1039, 2015.

[39] S. G. Manyam, S. Rathinam, S. Darbha, D. Casbeer, Y. Cao, and P. Chandler. Gps denied uav routing with communication constraints. *Journal of Intelligent & Robotic Systems*, 84(1-4):691–703, 2016.

[40] D. Marmanis, J. D. Wegner, S. Galliani, K. Schindler, M. Datcu, and U. Stilla. Semantic segmentation of aerial images with an ensemble of cnss. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2016*, 3:473–480, 2016.

[41] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. *arXiv preprint arXiv:1803.06815*, 2018.

[42] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[43] M. Müller, S. Urban, and B. Jutzi. Squeezeposenet: Image based pose regression with small convolutional neural networks for real time uas navigation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:49, 2017.

[44] T. Naseer and W. Burgard. Deep regression for monocular camera-based 6-dof global localization in outdoor environments. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 1525–1530. IEEE, 2017.

[45] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[46] S. Rady, A. Kandil, and E. Badreddin. A hybrid localization approach for uav in gps denied areas. In *System Integration (SII), 2011 IEEE/SICE International Symposium on*, pages 1269–1274. IEEE, 2011.

[47] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint arXiv:1505.04597*, 2015.

[48] F. Rottensteiner, G. Sohn, J. Jung, M. Gerke, C. Baillard, S. Benitez, and U. Breitkopf. The isprs benchmark on urban object classification and 3d building reconstruction. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 1(3):293–298, 2012.

[49] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.

[50] V. Santhanam, V. I. Morariu, and L. S. Davis. Generalized deep image to image regression. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[51] O. Saurer, G. Baatz, K. Köser, M. Pollefeys, et al. Image based geo-localization in the alps. *International Journal of Computer Vision*, 116(3):213–225, 2016.

[52] P. Shukla, S. Goel, P. Singh, and B. Lohani. Automatic geolocation of targets tracked by aerial imaging platforms using satellite imagery. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40(1):381, 2014.

[53] M. Siam and M. Elhelw. Enhanced target tracking in uav imagery with pn learning and structural constraints. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 586–593. IEEE, 2013.

[54] D.-G. Sim, R.-H. Park, R.-C. Kim, S. U. Lee, and I.-C. Kim. Integrated position estimation using aerial image sequences. *IEEE transactions on pattern analysis and machine intelligence*, 24(1):1–18, 2002.

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

[56] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34, 1948.

[57] J. Su, J. He, P. Cheng, and J. Chen. A stealthy gps spoofing strategy for manipulating the trajectory of an unmanned aerial vehicle. *IFAC-PapersOnLine*, 49(22):291–296, 2016.

[58] Y. Tian, C. Chen, and M. Shah. Cross-view image matching for geo-localization in urban environments. *arXiv preprint arXiv:1703.07815*, 2017.

[59] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Deep end2end voxel2voxel prediction. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, pages 402–409. IEEE, 2016.

[60] A. Viswanathan, B. R. Pires, and D. Huber. Vision based robot localization by ground to satellite matching in gps-denied situations. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 192–198. IEEE, 2014.

[61] S. Workman, R. Souvenir, and N. Jacobs. Wide-area image geolocalization with aerial reference imagery. In *Proceed-*

ings of the IEEE International Conference on Computer Vision, pages 3961–3969, 2015.

[62] A. Yol, B. Delabarre, A. Dame, J.-E. Dartois, and E. Marchand. Vision-based absolute localization for unmanned aerial vehicles. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3429–3434. IEEE, 2014.

[63] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs. Predicting ground-level scene layout from aerial imagery. *arXiv preprint arXiv:1612.02709*, 2016.

[64] Q. Zou, S. Huang, F. Lin, and M. Cong. Detection of gps spoofing based on uav model estimation. In *Industrial Electronics Society, IECON 2016-42nd Annual Conference of the IEEE*, pages 6097–6102. IEEE, 2016.