

Low Altitude UAV Video Stabilization Using Ground Plane Segmentation for Object Tracking

Deniz Kavzak Ufuktepe^a, Jaired Collins^a, Hadi AliAkbarpour^a, Priya Narayanan^b, Guna Seetharaman^c, and Kannappan Palaniappan^a

^aComputational Imaging and VisAnalysis Lab, Electrical Engineering and Computer Science Department, University of Missouri, Columbia, MO 65211 USA

^bIntelligent Perception Branch, CCDC Army Research Laboratory, Adelphi, MD 20783

^cAdvanced Computing Concepts, U.S. Naval Research Laboratory, Washington DC, 20375

ABSTRACT

Fast, efficient and robust algorithms are needed for real-time visual tracking that could run smoothly onboard airborne embedded systems. In order to efficiently use a motion-detection-based tracking algorithm on a raw image sequence captured by a moving platform, the relative motion between the camera and the ground plane must be stabilized first so that only the real moving objects in the scene could be detected. Using point correspondences to estimate the homography matrix between the frames is a common registration method used for the stabilization. In order to have a valid homography estimation, most of the point correspondences between the images have to lie on the same 3D plane (i.e. ground plane). However, when the scene includes multiple complex 3D structures (e.g. trees, cloudy sky, buildings) it becomes extremely challenging to satisfy this constraint, particularly for aerial images acquired from a low altitude flying drone with an oblique camera angle imaging a scene in which the ground plane is not the dominant object. In this work, we propose the Short-term Ground Plane-based Stabilization (SGPS) algorithm, a robust video stabilization algorithm that registers short video segments using the ground plane information to efficiently stabilize oblique aerial image sequences captured by low-altitude flying UAVs. Our experiments show satisfactory results when compared to registration without plane segmentation on the same type of raw videos.

Keywords: Video stabilization, low altitude UAV, registration, plane segmentation, homography estimation

1. INTRODUCTION

Real-time visual tracking on the aerial and surveillance imaging is an important task for image processing and computer vision. Since these imaging data are mostly needed for security demands, fast, efficient and robust algorithms are needed that could also run smoothly on the airborne (i.e. drone) embedded systems. Motion analysis is a powerful tool for visual tracking and computing, and the flux tensor¹—a robust motion-based object detection method—can be used for the real-time visual tracking to provide motion-based cues in addition to the appearance-based ones (e.g. YOLO²). In order to use any object motion detection on a raw image sequence captured by a moving platform (e.g. drone), the motion caused by the camera movement must be stabilized first.^{3–5} On the other hand, embedded systems have limited resources, so there are performance and energy efficiency considerations.⁶ Tracking algorithms usually have multiple steps in their pipeline, including stabilization itself like in an example traditional motion-based tracking pipeline shown in Figure 1. Running this pipeline requires sharing of embedded processing resources on an Unmanned Aerial Vehicle (UAV). To this end, we need a fast, efficient and robust stabilization module that minimizes the amount of resources used.

Using feature points to estimate the homography matrix between the frames is a simple registration method that can be used for stabilization. In order to have a good homography estimation, most of the feature points should lay on the same plane in the images. However, when the scene has complex structures with multiple 3D geometries, textures, multiple dominant planes, etc., it becomes very challenging to estimate a good homography.

Corresponding authors: Deniz Kavzak Ufuktepe, E-mail: deniz.kavzakufuktepe@mail.missouri.edu
Jaired Collins, E-mail: jairedcollins@mail.missouri.edu

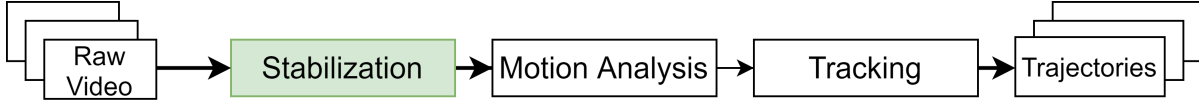


Figure 1. A traditional motion based tracking pipeline with video from a moving platform camera.

This is due to the fact that the underlying homography assumption, which is to be induced by a 3D plane in the scene, can be easily violated as a big percentage of the feature matches between the two frames can lie off the scene’s dominant ground plane. In presence of relatively low percentage of matching outliers (i.e. points that do not lie on the dominant plane), some statistical methods such as RANSAC can be used to filter out the outliers.⁷ However, for a general low-altitude UAV video of a complex scene using only these estimation models would not produce robust homography estimations, and more sophisticated methods have to be developed to deal with this problem.

In this work, we propose Short-term Ground Plane-based Stabilization (SGPS), a robust short-term video stabilization algorithm which allows motion detection methods like flux to efficiently identify moving objects. Proposed pipeline leverages the ground plane segmentation to mask the ground plane from the original frames. Particularly, PlaneRCNN,⁸ a deep learning based plane segmentation method, is used to get the plane segmentation in this work. Once the masked ground plane is found, feature based registration methods like a homography transformation can be used with higher success, since at least the majority of the points will lie on the same ground plane. An adaptive reference frame update approach is used to select reference frames on the go, and a group of frames in that corresponding group are transformed to their reference frame’s coordinate system. The reference update is decided by monitoring the amount of movement between the current frame and the last selected reference frame, together with the percentage of inlier feature points corresponding to the estimated homography between them. Plane segmentation is done for each reference frame only. Since the movement is smaller than a threshold in that group, the ground mask is reused until the next reference frame update. Our experiments show satisfactory results when compared to traditional registration methods without any ground plane segmentation on the same type of raw videos.

2. RELATED WORK

Unmanned aerial vehicles (UAVs) and drones can fly at various altitudes to collect video data. Unfortunately, the videos can be unstable due to gusts of wind, turbulence, or any other event that disturbs pose and trajectory. Therefore, as a pre-processing step, it is important to stabilize the video before using it in an algorithm.

Video stabilization typically starts with feature-based image registration, where the different sets of points in images are transformed into a single coordinate system. This can be achieved by estimating either a fundamental or homography matrix between images and then using it to transfer the pixels of one image to the other.⁹ The base image coordinate system is referred to as the reference frame and can be chosen using a fixed or adaptive interval by using a metric.¹⁰

Neel, et al.¹¹ has an implementation of video stabilization that’s focused on timelapses made from cameras with motion, as opposed to a stationary camera. Long videos may be shortened for many reasons, such as length, into a timelapse. Naïve timelapses suffer from jerky movements, so Neel et al. sought to fix this, calling a stabilized timelapse a hyperlapse. They achieve this by first estimating how well each frame can be aligned to its temporal neighbors, then finding an optimal path of frames that minimizes the frame-to-frame motion while matching the target frame rate, and then smoothing the camera path with the selected frames. While producing impressive results, a hyperlapse is only calculable after the sequence is taken, and it is not for real-time stabilization.

Real-time tracking is easier to compute from stabilized video. It is common to lose tracks of objects from camera jerk, so stabilization is a way to reduce the probability of a failure. Additionally, it is a difficult task to be done in real-time due to computational limits. There are many ways to stabilize video, but few are considering the computational burden and time to stabilize fast enough.

Most stabilization techniques track features over a sequence of frames, but an alternative and more robust way is to use optical flow. This can be computationally difficult and introduce extra execution time, which would

push the algorithm to not be real-time. However, Mitchel et al. created the Quotienting Impertinent Camera Kinematics (QuICK) motion model to recover 3D camera motion.¹² It is implemented and demonstrated to work with RGB-D videos, but it can work with any video where depth can be estimated. While robust, we aim to use monocular data—which does not encode depth data—obtained from mid-flying drones. There are algorithms that are capable of estimating depth but that would introduce more computations that are ultimately unnecessary. There are also some efficient 3D-enabled georegistration and video stabilization techniques that can be applied if the camera poses are available from the airborne sensors. In^{13–15} a method was proposed a method for global optimization of noisy camera poses. Aliakbarpour et al. in¹⁶ introduced a method to analytically compute homography matrices after refining the global and robust camera pose refinement.¹⁵ Another method which performs both global optimization and stabilization was recently proposed in.¹⁷

A 2D optical flow algorithm was made by Lim et al. Corner detectors are used to find the optical flow, then a motion estimation is created.¹⁸ A rigid or similarity transformation is returned from their function depending on an accuracy measurement. The method is extremely fast due to its ability to choose between lower order homographies, but its real-time performance is only achieved by the use of three concurrent threads.

Hardware based solutions have been created as well. An inertial measurement unit or jitter sensor can measure unintended movements that can be used to compensate the imagery for better stabilization.^{19–21} Pose can directly be used to improve the calculation of a homography or fundamental matrix. An orientation sensor may be used to better stabilize between large rotational variances.²² Additionally, GPS or barometric sensors can provide useful metadata to vastly increase air-to-ground image processing performance.²³ While the hardware-based solutions make stabilization easier, it also requires extra sensors that may incur cost or additional computational complexity. For this reason, we used a software-based approach.

3. SHORT-TERM VIDEO STABILIZATION ALGORITHM USING PlaneRCNN

Proposed short-term stabilization algorithm consists of three main parts that can be seen in Figure 2. The first part is the plane segmentation by PlaneRCNN⁸ on the selected reference frame to get the ground plane mask. The second part is the feature extraction and matching on the ground of the reference frame and the ground of the current frame, extracted using a reference frame mask. Finally, a homography estimation is done using the inlier feature points from the ground plane, that are further reduced by RANSAC, and the current frame is registered to the reference frame by using the estimated homography if it is not updated as a new reference frame.

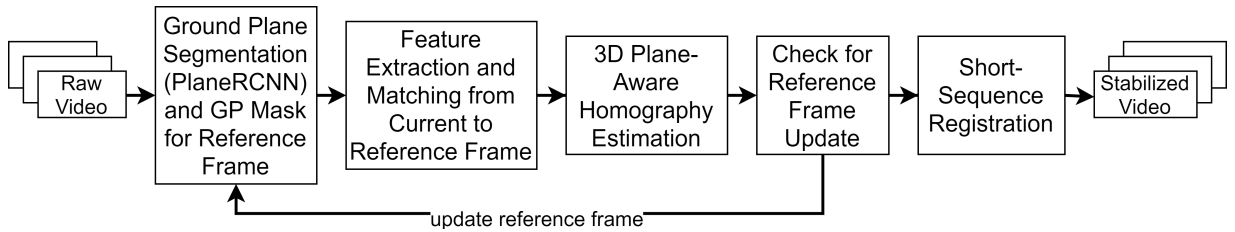


Figure 2. Proposed SGPS pipeline for short-term ground plane segmentation followed by image stabilization.

The reference frame is updated when one of the two update conditions occur. There is a displacement value calculated between each current frame and the last updated reference frame. Whenever the displacement is higher than the pre-defined threshold, or whenever the percentage of number of in-liers over all matched points are smaller than a pre-defined threshold, the reference frame is updated as the current frame. Hence, there are video fragments that are stabilized short-term in between each reference frame updates. The overall flow of the Ground Plane Segmentation-Based Stabilization Algorithm with the intermediate results for one of the experimented videos can be found in Figure 3. More information on ground plane segmentation and masking can be found in Section 3.1, feature extraction and matching are explained in Section 3.2 and homography estimation and stabilization is detailed in Section 3.3.

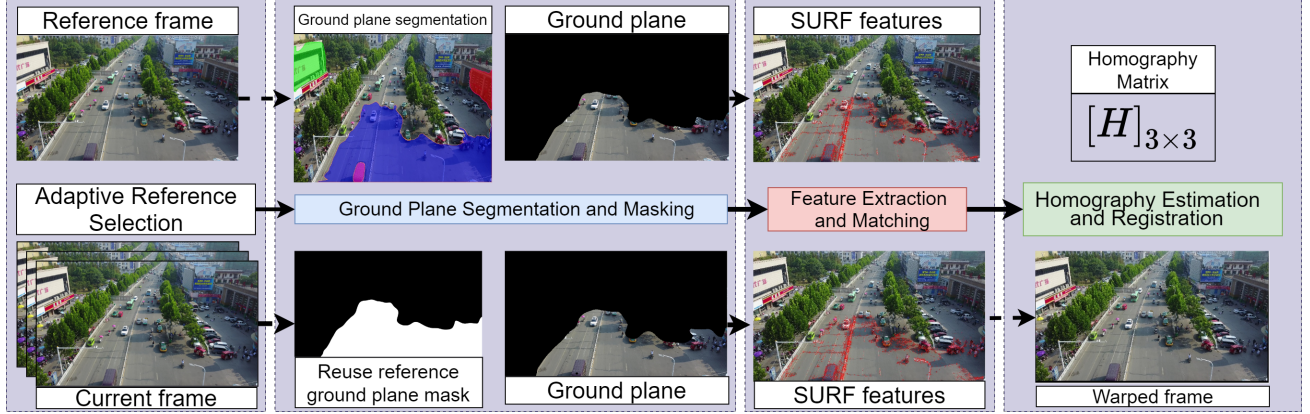


Figure 3. Ground Plane Segmentation-Based Stabilization algorithm flow with intermediate results of the video sequence *uav0000148_00840.s*.²⁴ From left to right: Segment planes on reference frame by PlaneRCNN and ground plane mask is taken on top row, ground plane is masked in both reference frame and the current frame using the reference frame ground plane mask, keypoints are extracted from masked ground image and descriptors computed, points are matched and a homography estimation is done. Current frame is transformed into reference frame coordinate space.

3.1 Ground Plane Segmentation and Masking

The first frame is selected as the initial reference frame and updated using an adaptive approach real-time, which is further explained in the Section 3.3.2. Plane segmentation results are taken by running PlaneRCNN⁸ on the reference frame, then the ground plane mask is extracted. The ground plane segmentation result image from PlaneRCNN is resized back to the original image size, since the resulting plane segmentation image is not the original image size. The biggest plane from the segmentation result is selected as the ground plane. The ground is masked from the reference image by using the ground mask obtained. As the new frames come, the same ground plane mask from the reference frame is used to get the ground image from the current frame. Both masked ground images of the reference and current frames are used in the following feature extraction and matching step.

3.1.1 PlaneRCNN

PlaneRCNN is a plane detection algorithm that is capable of extracting plane information within an image without specifying the number of planes which is often required in other approaches. It has three modules, a plane detection network, segmentation refinement network, and a warping loss module.⁸ Figure 4 shows the overall deep learning architecture of PlaneRCNN that is used to detect and reconstruct an arbitrary number of planar regions in a single RGB image. PlaneRCNN represents planes using three parameters - unit normal times the distance offset.

The plane detection network using Mask R-CNN,²⁵ where instead of object categories, the network only has two: planar or non-planar.⁸ The segmentation refinement network takes the output of the plane detection network and processes respective masks, normals, original images, and the union of all other masks for refinement. Finally, the warping loss module uses a nearby frame defined at 20 frames ahead and projects, unprojects, and transforms coordinate frames between them to increase consistency of reconstructed 3D planes. This is differentiable and trainable, so it is only used during training to boost geometric reconstruction accuracy. Because of its robustness, we chose PlaneRCNN to calculate our ground planes. Some example plane segmentation results from PlaneRCNN on images from different VisDrone datasets²⁴ can be seen in Figure 5.

3.2 Feature Extraction and Matching

Masked ground images of the reference and current frames are taken as inputs, together with the used mask. Feature extraction, description and matching methods are selected as desired with respect to the performance requirements and nature of the scenes in the video. In this work, two different type of feature extractors with

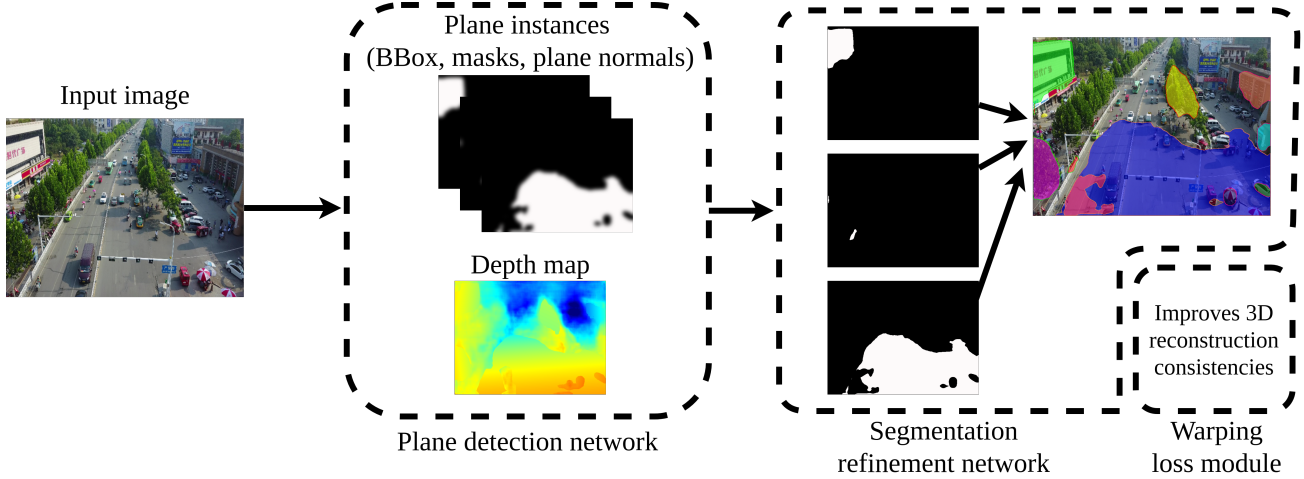


Figure 4. High-level PlaneRCNN deep learning architecture using a Mask R-CNN plane detection backbone for segmenting planar regions in a single image.

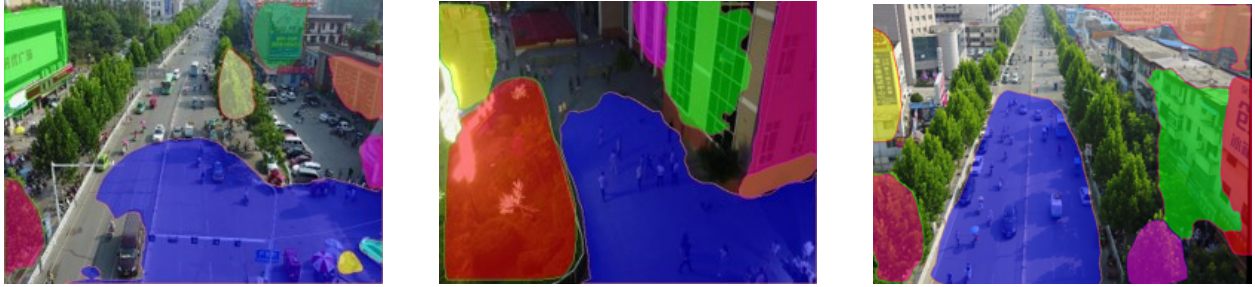


Figure 5. PlaneRCNN plane segmentation results for three sample images from the VisDrone dataset.

different characteristics have been chosen to be experimented with. First, ORB²⁶ is used since it is a binary descriptor and it is suitable for real-time applications.^{27–29} As a second experiment, Speeded Up Robust Features (SURF)³⁰ is used to see the differences caused by the selection of features between a corner based and a blob based one. Another point of comparison was intended to be the performance between ORB and SURF. In order to exclude the possible false feature points that would come from the ground mask corners in the ground image: First, the original ground mask is dilated, then the feature points are extracted from the dilated masked area from the image. Then, the feature points between the original mask and the dilated mask are excluded to discard those false non representative feature points. After the relevant points are found by excluding those points, descriptors are computed and feature points are matched. Matched points from reference image and the current image are given to the homography estimation and stabilization step as inputs.

3.3 Homography Estimation and Stabilization

Matched points from reference image and the current image are given to the homography estimation and stabilization step as inputs to estimate a homography transformation matrix from current frame to the reference frame. Random Sample Consensus (RANSAC) is used in the homography estimation to exclude further outliers and to have a more robust homography matrix that better represents the projection between the frames.⁷ Once the homography is estimated, the adaptive reference frame update rules are checked to decide whether the current frame will be transformed into the coordinate space of the reference frame, or it will be updated as a new reference frame. More details on the adaptive reference update rules can be found in the Section 3.3.2. If update rules do not indicate a need for update, the current frame is warped with respect to the reference frame to be registered and stabilized. If the update rules show that the current frame should be selected as a new reference frame, the reference frame is updated as the current frame, and the algorithm continues for the next incoming

frame. Warped frames in between reference frame updates create a short-term stabilized video segment that can be used for more accurate applications of motion based object tracking methods.

3.3.1 Homography Matrix Estimation

Homography matrix $H_{3 \times 3}$ can be used to represent the relationship between two images sharing the same planar surface, i.e. dominant plane. Generally, the relationship between two points taken from the current image to the reference image can be expressed as

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \quad (1)$$

where $[x \ y \ 1]^T$ is a point from the current image, $[x' \ y' \ 1]^T$ is the point from the reference image that will correspond to the point from the current image, s is a non-zero scaling factor and H is the homography matrix to be estimated. The last element in Eq. (1), h_{33} , can be set to one as the relation is up to a scale factor, resulting to the 8 degrees of freedom for H , which can be estimated using a minimum of 4 point correspondences (of a common 3D plane) between the two views. H can be estimated using a Direct Linear Transformation (DLT) estimation technique to minimize the following cost function:

$$E = \sum_i \left(x'_i - \frac{h_{11}x_i + h_{12}y_i + h_{13}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 + \left(y'_i - \frac{h_{21}x_i + h_{22}y_i + h_{23}}{h_{31}x_i + h_{32}y_i + h_{33}} \right)^2 \quad (2)$$

The widely used Random Sample Consensus (RANSAC) can be utilized to improve the accuracy of the homography matrix by eliminating outlier between those two images.⁷ Within this context, outliers are the feature correspondences that are either mismatched or do not represent a global 3D plane between the views. For a given number of iterations, RANSAC simply selects four random points from the putative matched points set that is being used for the homography estimation. Those four points are used to estimate a homography H' , and that H' is used to transform the rest of the matched points by checking the error measurement between them, in a similar manner to the error function E in Eq. (2) that is being used in homography estimation itself. If the distance between original matched point and the transformed point on the same image is smaller than a given threshold, that point is labeled as an inlier, if not, it is labeled as an outlier. In each iteration for each 4 points, the number of in-liers that are determined are counted, and after all iterations end, the four points with the maximum number of in-liers are selected as the best one. Finally, a new and final homography H is estimated by using the best four points and the corresponding in-liers with respect to those four points.

3.3.2 Adaptive Reference Frame Update

An adaptive reference frame update method is essential to accurately estimate the homographies needed to stabilize a short-term group of frames.^{9,10} When the frame-to-frame displacements become too large or the number of in-liers too few then matching becomes more difficult or unreliable and the reference frame should be updated as described in Algorithm 1 and 2. In order to decide when to update a reference frame which will create a stabilized group of frames, two rules were used.

First, the amount of movement in the scene between the reference frame and current frame should be less than a threshold. The displacement or 2D translation $d = [d_x, d_y]$ is estimated between the last selected reference frame and the current frame. The displacement value consists of the estimated movement in both x - and y -axis. Once the homography H is estimated between the current frame and reference frame, 10 points from the original matched points are selected randomly. Those 10 points are transformed using H into the coordinate space of the reference frame. The mean of the distance between all transformed points and the corresponding selected points are taken on both axes as the displacement d . The mean distance on the x -axis is called d_x and the y -axis is called d_y . A pre-defined factor α is used to compare d_x and d_y values with $\alpha * width$ and $\alpha * height$ respectively, where width and height are the size of the image. If any of the displacement values are greater than the compared threshold, it means the movement is greater than it is desired, therefore the reference frame is updated as the current frame.

Second, in addition to the displacement rule, the ratio of the number of in-liers from the homography estimation process over the number of matched points is taken as the quality value. If the quality value of the current homography estimation H is smaller than a pre-defined threshold β , it means the number of in-liers that were fit on the homography matrix are a smaller portion of the matched points—smaller than desired. Therefore the reference frame is updated as the current frame. If any of the rules apply and the reference frame is updated, the current frame is not warped, is used directly in the registered output, and a new stabilized segment is started. If any of the rules do not apply and the reference frame is not updated, the same reference frame is continued to be used for the next frame, while the current frame is warped into the coordinate space of the reference frame using H , and the stabilized segment continues.

Algorithm 1 Adaptive Reference Frame Update

```

1: procedure UPDATE(reference, current, GPmatches, in-liers,  $H$ ,  $\alpha$ ,  $\beta$ )
2:   Input: reference: last selected reference frame
             current: current frame
             GPmatches: ground plane matched points between current frame and reference frame
             in-liers: in-liers from matched points used in the homography estimation
              $H$ : estimated homography matrix from current frame to reference frame
              $\alpha$ : factor for the displacement thresholding
              $\beta$ : factor for the in-lier thresholding
   Output: reference: updated reference frame
             warped: warped image result for current frame
3:    $width, height \leftarrow size(reference)$ 
4:    $d = (d_x, d_y) \leftarrow DISPLACEMENT(in-liers, H)$ 
5:    $quality \leftarrow (size(in-liers) / size(GPmatches))$ 
6:   if  $d_x > \alpha * width$  or  $d_y > \alpha * height$  or  $quality < \beta$  then                                 $\triangleright$  check update rules
7:      $reference \leftarrow current$                                                                  $\triangleright$  update reference frame as current
8:      $warped \leftarrow current$                                                                  $\triangleright$  no warping, use current directly
9:   else                                                                 $\triangleright$  no update needed for reference frame
10:     $warped \leftarrow WARP(current, H)$                  $\triangleright$  warp current image into reference frame coordinate system
11:  return reference, warped

```

Algorithm 2 Displacement Estimation

```

1: procedure DISPLACEMENT(in-liers,  $H$ )
2:   Input: in-liers: in-liers from matched points used in the homography estimation
              $H$ : estimated homography matrix from current frame to reference frame
   Output:  $d$ : estimated displacement
3:    $points \leftarrow \{\}$                                  $\triangleright$  random points from current ground plane matched points
4:    $transformed\_points \leftarrow \{\}$                      $\triangleright$  transformed points from current to reference coordinate space
5:   for  $i \leftarrow 1, 10$  do
6:      $m = (m_1, m_2) \leftarrow RAND(in-liers)$            $\triangleright$  random matched pair of in-liers:  $m_1$  from current,  $m_2$  from
reference
7:     add  $m_1$  to  $points$                                  $\triangleright$  add  $m_1$  to points
8:      $pt \leftarrow H * m_1$                                  $\triangleright$  transform point from current to reference coordinate space
9:     add  $pt$  to  $transformed\_points$                      $\triangleright$  add transformed point to transformed_points
10:   $d = (d_x, d_y) \leftarrow MEAN(transformed\_points - points)$   $\triangleright$  get mean difference of all
11:  return  $d$ 

```

4. EXPERIMENTAL RESULTS

Proposed algorithm has been experimented on two videos from the single object tracking training datasets in the VisDrone 2019 challenge, *uav0000070_02088_s* and *uav0000148_00840_s*.²⁴ For both videos, the registration based stabilization algorithm using our adaptive reference frame update scheme have been experimented. Both with ground plane segmentation and without ground plane segmentation cases have been experimented to see stabilization improvement when ground plane segmentation information is used. All experiments were repeated using ORB and SURF features for feature selection comparison to see the affect of different feature detection and description method characteristics. For all experiments, brute force matching with L1 Norm has been used in the feature matching phase.

There are several points that have been addressed for better time performance measures and less resource usage. For the feature extraction and matching parts of the algorithm, the main performance gain is from extracting feature points from only the ground plane, instead of the whole image. This reduction is beneficial in terms of both performance, efficiency and resource utilization. Moreover, the feature points were extracted from the ground plane of the reference frame once and reused until the next reference frame update for further performance improvement. Feature points extracted from whole image and from the masked ground image for example selected frames from both experimented videos can be seen in Figure 6. It can be seen that the huge amount of feature points are excluded without even being extracted by using ground plane segmentation.



Figure 6. Keypoints extracted by SURF from a UAV captured frame from the video sequence *uav0000148_00840_s* and *uav0000070_02088_s*²⁴ respectively. Left: without plane segmentation. Right: using ground plane segmentation.

In the experiments, same adaptive reference frame update rules are applied and the same feature-based registration is done by homography estimation for both cases where there is a ground plane segmentation and not. Therefore, same experiments are applied to the full image with all extracted features, and then using the masked features extracted from the ground plane region after it is segmented. In this comparison, it was seen that ground plane segmentation improved results with both ORB and SURF features. Examples of consecutive warped frames without and with ground plane segmentation for both of the experimented videos can be found in Figure 7. In this figure, there are two image pairs where each pair represent a stabilization result for consecutive frames. The top ones for both videos are without and the bottom ones for both videos are with ground plane

segmentation. Here, it can be seen that the warping, i.e. stabilization result is more accurate in the case where we apply ground plane segmentation, when both consecutive frames are transformed into a common coordinate space. The selected threshold for RANSAC is 3 pixels in all experiments. In the adaptive reference frame update, the displacement threshold for displacement $\alpha = 1/3$, leading on the threshold on x-axis as $1/3 * width$, and on the y-axis as $1/3 * height$, where width and height are the size of the image. The percentage threshold for the number of in-liers over the number of matched points is selected to be $\beta = 40\%$.

Another point of comparison was using two different feature point extraction and description methods with different characteristics to see the affect of both on the stabilization results. All experiments are repeated with ORB and SURF methods while all the other parameters were fixed. In terms of stabilization success, experiments on the selected videos, using SURF gave slightly better results, whereas ORB was able to find less number of feature points from the same ground plane of the scene. In terms of time performance, ORB performed much faster, and SURF is not as suitable as ORB in real-time applications performance-wise. Some experimental measurement results—done on Hardware configurations: 12 GB RAM, Intel Core i7-6700HQ CPU @ 2.60GHz, Nvidia GeForce GTX 950M—can be found in Table 1. In these experiments, when ORB features are used, measured frame per second values change approximately in the range 4 to 8, depending on the image size, the nature of the video, coverage area of the video, number of reference frame updates, etc. However, when SURF is used in the measurement experiments, the resulting fps is far from being applicable in real-time scenarios. More observations on the time performance, the overhead of running PlaneRCNN on only reference frames did not exceed the performance gained from extracting less features instead of the whole image. As a result, it was seen that the approach of ground plane segmentation based registration for short term video stabilization can be optimised further and can be used in real time applications using resources efficiently.

Dataset	ORB					SURF
	070_02088_s	305_00000_v	182_00000_v	148_00840_s	268_05773_v	148_00840_s
#Frames	217	184	363	807	978	807
Image size (w,h)	1904×1071	1904×1071	1344×756	2688×1512	3840×2160	2688×1512
#Ref Frames	5	5	16	33	3	29
Time (mm:ss)	00:28	00:27	00:54	02:56	04:03	49:57
FPS	7.8	6.8	6.7	4.6	4.0	0.27

Table 1. The frame rate performance (last row) is a function of number of reference frames (calls to PlaneRCNN), image size and the ground plane mask size. Time measurement results of the short-term stabilization algorithm on 5 datasets from VisDrone 2019.²⁴ Hardware configurations: 12 GB RAM, Intel Core i7-6700HQ CPU @ 2.60GHz, Nvidia GeForce GTX 950M. Experiment results of the datasets using ORB features, and result of one of the datasets with SURF features. Dataset name—start with uav0000—, Number of frames in the dataset, Image size of the video frames, number of updated reference frames, time measurement of the algorithm, corresponding frame per second results.

Moreover, in order to compare the effects of different feature extraction and detection methods, the reference frame update rate has been compared. Since SURF was able to extract more features than ORB, the update rate was slightly lower in average. Meaning there were longer stabilized sequences in the experiments with SURF, but not with a significant difference compared to the length achieved by ORB. In some of the other initial experiments that have been done on a video where the camera movement coverage was a smaller area, it was seen that the same reference frame could be used for more than 100 frames. However, in the experiments from VisDrone dataset, since the camera movement was much higher in a direction in addition to the camera jitters at the same time, the update rate was much higher with both ORB and SURF. Meaning the stabilized segments were shorter, with 20 frames in average. Still, it was seen that proposed algorithm can be used to improve the applicability of motion based tracking methods since it provides a good amount of short-term stabilized video segments.

5. CONCLUSIONS AND FUTURE WORK

In this work, the experiments showed that our ground plane segmentation-based short-term stabilization algorithm gave promising results for low flying UAV drone videos. Using ground plane segmentation improves homography estimation and reduces the number of outlier frames after the registration. Better estimation of the

homography, together with the adaptive reference frame update, results in a better registration, hence a better short-term video stabilization. These short segments, 20 to 100 in average, of stabilized videos can be used for motion-based tracking algorithms. The parameters of the adaptive reference frame update can be tweaked to have longer but less accurate stabilized segments, or to have shorter with higher accuracy stabilized segments, with respect to the needs.

Moreover, plane segmentation improves the efficiency of feature extraction computation, since the extraction area is reduced to only ground plane instead of the whole image. The proposed algorithm can be used with any feature extraction, description and matching methods, selected according to the nature of the scene, time and performance constraints. Using ORB features is fast and can be used to achieve real-time stabilization, whereas SURF features give slightly more accurate stabilization results but worse for real-time stabilization under strict time constraints. Proposed plane segmentation based registration for short-term video stabilization algorithm using ORB features gives promising results in time performance measurements, therefore can be further optimised to achieve higher fps values for real-time applications.

For extending and improving this work, there are some challenges that need to be tackled and solved. One of the challenges is the smaller objects on the ground plane like pedestrians and cars that contain lots of feature points but are deceiving for the homography estimation process, especially if those small objects are moving. Many wrong matches are being made and the registration is affected by them. In order to detect and exclude those small objects, an object detection method like YOLO² can be used to further refine the ground plane feature points, for it has already been shown that it can run on UAV hardware.³¹ Moreover, the tracker information can be used as a feedback to exclude more feature points that do not belong to the stationary ground to improve the online stabilization.

Another point of view that needs to be addressed is the ground plane selection. To make this algorithm more widely applicable and robust for different datasets, we can use PlaneRCNN’s plane outputs for the whole scene. This can be used to determine and select the ground plane more intelligently, but with the added challenge of understanding the relative coordinates enough to find the ground plane. Moreover, if the camera intrinsic parameters can be provided, the normals of the planes can be estimated with higher accuracy, leading to a better ground plane selection. On the other hand, the dominant plane in the scene might not be the ground plane. Hence, the correct plane selection from the plane segmentation should be investigated further to make this algorithm applicable for a wider range of videos with different characteristics.

Finally, other plane segmentation methods can be investigated and used in the first step of the algorithm. Since both PlaneRCNN and Mask RCNN—that is used in PlaneRCNN—are trained on data with a much different nature, like indoors and close up datasets, one or both can be re-trained with low flying UAV datasets to have much better ground segmentation, which would lead to a more accurate and robust stabilization for longer segments.

ACKNOWLEDGMENTS

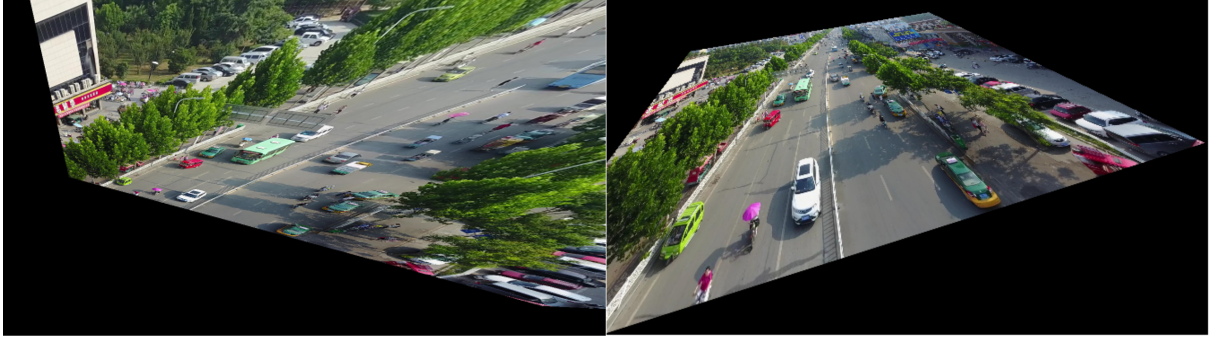
This work was partially supported by awards from U.S. Army Research Laboratory W911NF-1820285 and Army Research Office DURIP W911NF-1910181. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the U.S. Government or agency thereof.

REFERENCES

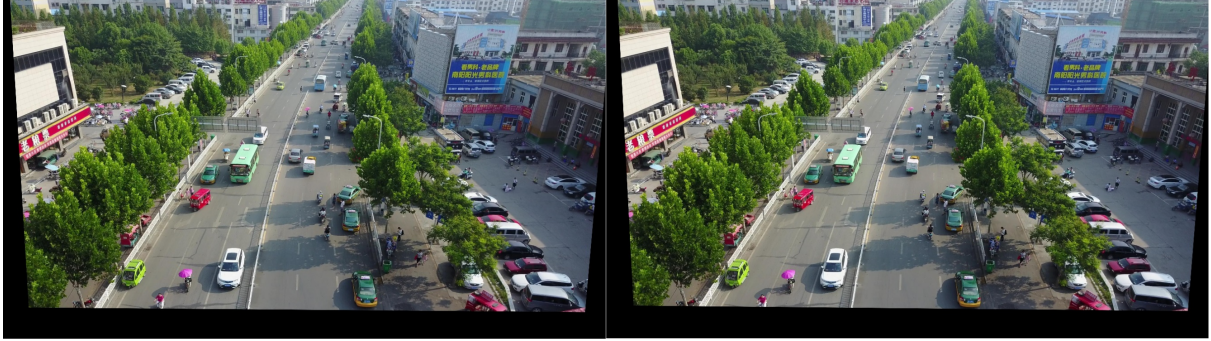
- [1] Wang, R., Bunyak, F., Seetharaman, G., and Palaniappan, K., “Static and moving object detection using flux tensor with split gaussian models,” in *[IEEE Conf. on Computer Vision and Pattern Recognition Workshops]*, 414–418 (2014).
- [2] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., “You only look once: Unified, real-time object detection,” in *[IEEE Conf. on Computer Vision and Pattern Recognition]*, 779–788 (2016).
- [3] Al-Shakarji, N., Bunyak, F., Aliakbarpour, H., Seetharaman, G., and Palaniappan, K., “Multi-cue vehicle detection for semantic video compression in georegistered aerial videos,” in *[IEEE Conf. on Computer Vision and Pattern Recognition Workshops]*, 56–65 (2019).

- [4] Palaniappan, K., Poostchi, M., Aliakbarpour, H., Viguier, R., Fraser, J., Bunyak, F., Basharat, A., Sudarth, S., Blasch, E., Rao, R. M., et al., “Moving object detection for vehicle tracking in wide area motion imagery using 4D filtering,” in [*Int. Conf. on Pattern Recognition*], 2830–2835 (2016).
- [5] Poostchi, M., Aliakbarpour, H., Viguier, R., Bunyak, F., Palaniappan, K., and Seetharaman, G., “Semantic depth map fusion for moving vehicle detection in aerial video,” in [*IEEE Conf. on Computer Vision and Pattern Recognition Workshops*], 32–40 (2016).
- [6] Teters, E., AliAkbarpour, H., Palaniappan, K., and Seetharaman, G., “Real-time geoprojection and stabilization on airborne GPU-enabled embedded systems,” in [*SPIE Geospatial Informatics, Motion Imagery, and Network Analytics VIII*], **10645**, 106450H (2018).
- [7] Fischler, M. A. and Bolles, R. C., “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM* **24**, 381–395 (June 1981).
- [8] Liu, C., Kim, K., Gu, J., Furukawa, Y., and Kautz, J., “PlaneRCNN: 3D plane detection and reconstruction from a single image,” in [*IEEE Conf. on Computer Vision and Pattern Recognition*], 4450–4459 (2019).
- [9] Aktar, R., AliAkbarpour, H., Bunyak, F., Kazic, T., Seetharaman, G., and Palaniappan, K., “Geospatial content summarization of UAV aerial imagery using mosaicking,” in [*SPIE Geospatial Informatics, Motion Imagery, and Network Analytics VIII*], **10645**, 106450I (Apr 2018).
- [10] Aktar, R., Prasath, V. B. S., Aliakbarpour, H., Sampathkumar, U., Seetharaman, G., and Palaniappan, K., “Video haze removal and Poisson blending based mini-mosaics for wide area motion imagery,” *IEEE Applied Imagery Pattern Recognition Workshop*, 1–7 (Oct 2016).
- [11] Joshi, N., Kienzle, W., Toelle, M., Uyttendaele, M., and Cohen, M. F., “Real-time hyperlapse creation via optimal frame selection,” *ACM Transactions on Graphics* **34**(4), 1–9 (2015).
- [12] Mitchel, T. W., Wulker, C., Kim, J. S., Ruan, S., and Chirikjian, G. S., “Quotienting impertinent camera kinematics for 3D video stabilization,” *IEEE Int. Conf. on Computer Vision Workshop* (2019).
- [13] AliAkbarpour, H., Palaniappan, K., and Seetharaman, G., “Fast structure from motion for sequential and wide area motion imagery,” in [*Int. Conf. on Computer Vision Workshop*], 1086–1093 (Dec 2015).
- [14] AliAkbarpour, H., Seetharaman, G., and Palaniappan, K., “Method for fast camera pose refinement for wide area motion imagery,” (Oct. 8 2019). US Patent 10,438,366.
- [15] Aliakbarpour, H., Palaniappan, K., and Seetharaman, G., “Robust camera pose refinement and rapid SfM for multiview aerial imagery—without RANSAC,” *IEEE Geoscience and Remote Sensing Letters* **12**, 2203–2207 (Nov 2015).
- [16] AliAkbarpour, H., Palaniappan, K., and Seetharaman, G., “Parallax-tolerant aerial image georegistration and efficient camera pose refinement—without piecewise homographies,” *IEEE Transactions on Geoscience and Remote Sensing* **55**(8), 4618–4637 (2017).
- [17] Aliakbarpour, H., Palaniappan, K., and Seetharaman, G., “Stabilization of airborne video using sensor exterior orientation with analytical homography modeling,” in [*Machine Vision and Navigation*], 579–595, Springer (2020).
- [18] Lim, A., Ramesh, B., Yang, Y., Xiang, C., Gao, Z., and Lin, F., “Real-time optical flow-based video stabilization for unmanned aerial vehicles,” *Journal of Real-Time Image Processing* **16**(6), 1975–1985 (2017).
- [19] Odelga, M., Kochanek, N., and Bühlhoff, H. H., “Efficient real-time video stabilization for uavs using only IMU data,” in [*IEEE on Research, Education and Development of Unmanned Aerial Systems*], 210–215 (2017).
- [20] Raut, S., Shimasaki, K., Singh, S., Takaki, T., and Ishii, I., “Real-time high-resolution video stabilization using high-frame-rate jitter sensing,” *ROBOMECH Journal* **6**(1) (2019).
- [21] Ramachandran, M. and Chellappa, R., “Stabilization and mosaicing of airborne videos,” in [*Int. Conf. on Image Processing*], 345–348 (2006).
- [22] Scaramuzza, D. and Fraundorfer, F., “Visual odometry [tutorial],” *IEEE Robotics & Automation Magazine* **18**(4), 80–92 (2011).
- [23] Wu, Z., Suresh, K., Narayanan, P., Xu, H., Kwon, H., and Wang, Z., “Delving into robust object detection from unmanned aerial vehicles: A deep nuisance disentanglement approach,” in [*IEEE Int. Conf. on Computer Vision*], (October 2019).

- [24] Zhu, P., Wen, L., Bian, X., Ling, H., and Hu, Q., “Vision meets drones: A challenge,” *arXiv preprint arXiv:1804.07437* (2018).
- [25] He, K., Gkioxari, G., Dollar, P., and Girshick, R., “Mask R-CNN,” in [*IEEE Int. Conf. on Computer Vision*], (Oct 2017).
- [26] Rublee, E., Rabaud, V., Konolige, K., and Bradski, G., “ORB: An efficient alternative to sift or surf,” in [*IEEE Int. Conf. on Computer Vision*], 2564–2571 (2011).
- [27] Mur-Artal, R., Montiel, J. M. M., and uan D. Tardós, “ORB-SLAM: A versatile and accurate monocular SLAM system,” *IEEE Transactions on Robotics* **31**(5), 1147–1163 (2015).
- [28] Mur-Artal, R. and Tardós, J. D., “ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras,” *IEEE Transaction on Robotics* **33**(5), 1255–1262 (2017).
- [29] Žak, B. and Hożyń, S., “Local image features matching for real-time seabed tracking applications,” *Journal of Marine Engineering & Technology* **16**(4), 273–282 (2017).
- [30] Bay, H., Tuytelaars, T., and Van Gool, L., “SURF: Speeded up robust features,” in [*European Conf. on Computer Vision*], 404–417, Springer (2006).
- [31] Narayanan, P., Borel-Donohue, C., Lee, H., Kwon, H., and Rao, R., “A real-time object detection framework for aerial imagery using deep neural networks and synthetic training images,” in [*SPIE Signal Processing, Sensor/Information Fusion, and Target Recognition XXVII*], Kadar, I., ed., **10646**, 25–33 (2018).



(a) Two consecutive frames registered without ground plane (GP) estimation



(b) Two consecutive frames registered with single GP estimation (the proposed method)



(c) Two consecutive frames registered without GP estimation



(d) Two consecutive frames registered with single GP estimation (the proposed method)

Figure 7. Registration of two consecutive frames in two VisDrone²⁴ video sequence datasets: *uav0000148_00840_s* (the first two rows) and *uav0000070_02088_s* (the last two rows). The results without using ground plane segmentation are presented in (a) and (c) for the two datasets, respectively. The results on the same datasets using our proposed method (GP) are presented in (b) and (d). The strong misalignment between the two adjacent frames is obvious in (a) and (c). This is because all the features shown in Fig. 6 are used for homography estimation, where many of them do not belong to the ground plane, causing confusion for the RANSAC homography. In contrast, our proposed method managed to efficiently register/align the frames by exploiting the 3D plane segmentation, as depicted in (b) and (d).