iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 5 **Issue:** III **Month of publication:** March 2017

DOI: http://doi.org/10.22214/ijraset.2017.3021

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

Volume 5 Issue III, March 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Tracking and Map Formatting of Road Using Unmanned Ground Vehicle

B. Suresh Kumar[1], G. Kadhirve[2], S. Balakrishnan[3]
[1,2,3]*Assistant Professor, Department of ECE, The Kavery Engineering College, Mecheri, Tamilnadu*

Abstract: ***An unmanned ground vehicle (UGV) has many applications in a variety of fields. Detection and tracking of a specific road in UGV videos play an important role in automatic UGV navigation, traffic monitoring, and is very helpful for constructing road networks for modeling and simulation. In this paper, an efficient road detection and tracking framework in UGV videos is proposed. In particular, a graph-cut–based detection approach is given to accurately extract a specified road region during the initialization stage and in the middle of tracking process, and a fast homography-based tracking scheme is developed to automatically tracking road areas. The high efficiency of our framework is attributed to two aspects: the road detection is performed only when needed and most work in locating the road is rapidly done via very fast homography-based tracking. Experiments are conducted on UGV videos of real road scenes we captured and downloaded from the Internet. The astonishing results indicate the effectiveness of our proposed work, with the precision of 98.4% and 1046 × 595 videos on average with processing 34 frames per second.***

Index Terms: Graph Cut algorithm, homography, road detection, road tracking, unmanned aerial vehicle (UAV).

I. INTRODUCTION

Unmanned ground vehicles (UGVs) have been widely used in many fields, particularly in transportation. The major applications include security surveillance, traffic monitoring, inspection of road construction, and survey of traffic, river, coastline, pipeline, etc. Relevant research can be traced back to the 2000s in the transportation departments of the Ohio [1], Florida [2], Georgia [3], and California [4] states within the United States. They use UGVs in autonomous navigation to follow roads/rivers, oil-gas pipeline inspection, and traffic parameters measurements. UAVs equipped with cameras are viewed as a low-cost platform that can provide efficient data acquisition methods for intelligent transport systems. With the increasing vehicles usage and their traffic management demands. this kind of platform becomes attractively popular. Conventional traffic data collection [5] relying on fixed infrastructure is only limited to a local region and, thus, it is expensive and labor intensive to monitor traffic activities across broad areas. In comparison, UAV has advantages, including:

There is a low cost to monitor over long distances; it is flexible for flying across broad spatial and temporal scales; and it is capable of carrying various types of sensors to collect abundant data. Detection in deep about road areas can provide users the regions of interest for further navigation, detection and data collection procedures, benefiting their efficiencies and accuracies.

In the previous works of road detection and tracking, most approaches use the color (texture) and/or structure (geometry) properties of roads. Among them, the combination of road color and boundary information have achieved more robust and ac-curate results than using only one of them in road detection, as shown in the work [6], [7]. Therefore, we are paying a note of using both types of information. Because real time is required in many UAV-based applications, our major target is how to effectively combine both types of information for road detection/ tracking in an efficient way. There are two rules that make one integrated framework efficient. First, each and every component of the framework should be fast. Second, if one component is faster than the others in achieving the same purpose, fastest component should be prioritized as much as possible.

we follow the above mentioned two rules to make the framework fast. our framework includes two components: road detection, road tracking. In road detection, we utilize the GraphCut algorithm [8] considering its efficiency and powerful segmentation performance in 2-D color images. In road tracking, we outcomes a fast road tracking approach. There are two facts that deviates us to implement road tracking. First, although GraphCut is very efficient, it still cannot achieve a real-time performance when the UAV image resolution is high enough (as in our work), and performing road detection frame by frame is not time efficient. Second, road appearance usually does not abruptly change in video; therefore, road tracking can make full use of continuous spatial–temporal information of roads in videos and thus can quickly infer road areas from previous results.

In road tracking, we track the road border structure between two consecutive frames. In a computer vision society, most developed tracking techniques, such as meanshift [9], particle filter [10], and optical flow [11], are appearance-based methods. They are for

pedestrian, where objects share common features. Although some contour-based methods, such as snake [12] or curve fitting [13] approaches, have shown promising road detection and tracking performance for applications to unmanned ground vehicles (UGV), they largely depend on the extracted road border (or markings) and vanishing points of the road, and might not be easily adaptable to UAV applications because the road boundary or markings are usually not salient enough to be detected due to the altitude of UAVs. In addition, these approaches are too computationally difficult to be used at real time.

Because of the aforementioned difficulties in adapting existent tracking approaches to track a road in UAV videos, we develop a novel tracking technique based on homography alignment. Homography is a method of transformation that can be used to align one image plane to other when the moving camera is capturing images of a plane scene. Generally, the road region in our interest can be well approximated by a plane, and therefore, homography can be applicable to our images. As mentioned, we aim at making our framework more efficient. We thus develop a fast homography estimation approach for road tracking, where the efficiency in homography estimation is attributed to three factors: (1) the FAST corner detector [14] is used to find key points in each road frame. (2) The Kanade–Lucas–Tomasi (KLT) tracker [15] is applied to establish a correspondence between the two sets of FAST corners in two consecutive frames. (3) A context-aware homography estimation approach is given where the corresponding FAST corners in the road neighbours are used with random sample consensus (RANSAC) estimator.

Based on our homography-alignment scheme, if we know where the road area is in a previous frame, we can quickly locate the corresponding road region in current frame. Fig. 1 illustrates the idea of road tracking based on homography alignment, where the road region in (a) captured at time t is known *a priori* and the corresponding road region in frame $t+1$ is tracked through mapping the road of (a) to (b) based on the estimated homography. In a similar way, we can also track the corresponding road area in frame $t+2$ based on the located road region in frame $t+1$, and so on so forth.

To get through this situation, we propose an online GraphCut scheme to detect the road area in the non-overlapping part. Since only a very small portion of the image is covered by non-overlapping part, the detection of road area in the non-overlapping part is very fast. Homography-alignment-based approach is much faster than GraphCut in location of the same road area. Therefore, the use of fast homography-based approach to track the road region in each frame can save considerable computation than mere use of GraphCut segmentation.

The contributions of this paper are listed as follows: (1) We propose a real-time framework for the detection and tracking of a specific road in low- and mid-altitude UAV videos. To our best knowledge, this is the first work that has ever been proposed to introduce a tracking technique to speed up the localization of road in UAV videos. (2) Mixture of technologies involving a fast homography-alignment-based tracking and an online GraphCut detection in an effective way to locate road region in each frame. The fast homography-alignment approach, which is context aware, is given to track most part of the road region and the online GraphCut detection to detect the rest trivial road area in every frame. (3) To correct the tracking error caused by the drift problem in successive homography alignment, we give an online drift-correction scheme, where a comparison is first made between the color distribution of the tracked road area in current frame and the one obtained from the tracked road regions in some previous frames, and a decision can be made based on the comparison whether it is necessary to switch from tracking to detection. Apart from normal road detection and tracking, our technique is applicable to river, pipeline, coastline detection and tracking of UAV videos.

Section II reviews some most related works to our technique. Section III provides the introduction of the hardware configuration. Section IV describes in details on how to use an offline GraphCut to initialize the road region and how to adjust GraphCut adaptively for road detection in the middle of tracking and drift-error correction. The elaboration of proposed fast homography-alignment-based road tracking technique is presented in Section V. Section VI presents bulk of experimental results in qualitative and quantitative ways, which gets concluded in Section VII.

II. RELATED WORK

Road detection and tracking in UAVs, particularly low- and mid-altitude UAVs is our focus in this paper, which can be used for autonomous navigation [4], [16], inspection [17], [18], traffic surveillance and monitoring [1], [20], [21]. A monocular color camera is equipped in this area, where UAVs usually are flying up to 500 m. The camera captures each vehicle clearly on the ground and contains large spatial view on traffic areas. The other research in UAV-based road detection uses high-altitude UAVs [22]–[24], which identify road network, including many junctions and roundabouts from an image. High-resolution cameras are utilized in the greater altitude UAV applications, where cameras is kept 1000 m away from ground. The third type of road detection works use ego-vehicles with onboard cameras with driver assistance systems or UGVs autonomous navigation systems. A substantial amount of work [7], [25]–[29] have been done in this area. Since the focus of this paper is on road detection and tracking using low-/mid-

Volume 5 Issue III, March 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

altitude UAVs, we only give a review of the most related works in this area.

In general, region color distributions or/and boundary structures are probably the most important information utilized for road detection. In [4], they proposed to learn road color distributions using Gaussian mixture models (GMMs) from given sample images, and then determine road pixels in each frame by checking the probabilities of pixels that fit the GMMs. Learning of both color and gradient information from a sample image is proposed in [17].

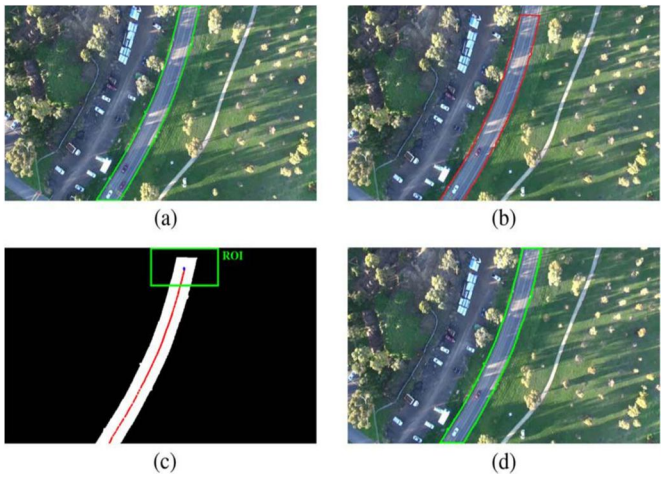


Fig.1.

Road tracking by homography , where the two frames (a) and (b) are not consecutive and they are selected to illustrate the clear gap. (a) Road detection result, where the contour of the road region is shown by green. (b) Result of road alignment by homography transformation. (c) Road median axis and the predicted region of interest (ROI). (d) Road detection in ROI and tracking result.

Gaussian and gamma distributions are used to represent information of color and gradient models. In [16], they learn structures from a sample image. Vanishing points are calculated by detecting pairs of line segments, and used to rectify the image in order to obtain rectified horizontal scans. Road boundaries are then identified through finding large intensity changes in the cross-section profile of horizontal scan. In [30], the clustering technique based on prior hue and texture information is used to classify each image pixel into target and background, and then boundary lines are fitted to refine the desired region. In [18], [19], a simple intensity thresholding technique is used to obtain initial road regions, followed by refinements of local line segment detections, where the assumption is that roads intensities are very different from neighborhood regions and roads can be approximated locally by linear line segments. Note that control algorithms of steering UAV in real time are also developed in [4], [16], which are not in the scope of this paper. Our objective is a fast real-time road tracking approach in UAV videos. Our future work will investigate how to use the tracked road information for parsing scenes in UAV videos.

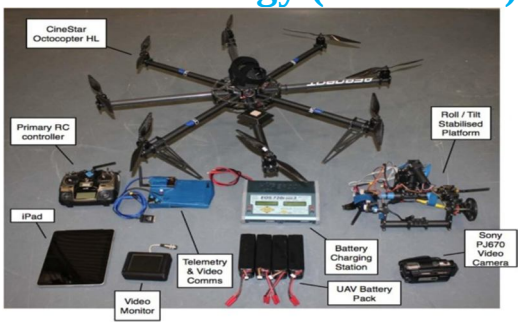
III. HARDWARE CONFIGURATION

The UAV used in this paper is configured as shown in Fig. 2, which includes several components: batteries and chargers, octo-copter, ground monitors and controllers, a camera and stabilization support. The octocopter is equipped with a global positioning system (GPS) and an altitude controller, which can fly up to 400 m within a 250 m radius. The fly can be autonomous by aid of programmed GPS waypoints or interactively by remote controls from users on the ground. The camera is carried at the bottom of the UAV, which is full HD in the progressive mode. The UAV is capable of tilting the camera to make it look straight down or with arbitrary tilted angles with regard to the ground. Some UAV images are shown which belongs to different days and at different places with varying altitudes. The road colors vary significantly, as well as the background

IV. ROAD DETECTION USING GRAPHCUT

First, we introduce a GraphCut-based road detection method, where the GMMs are used to model image color distributions, and structure tensors are employed to capture image edge features. To accelerate the performance, we perform road detection on downsampled images as in [7].

Volume 5 Issue III, March 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)



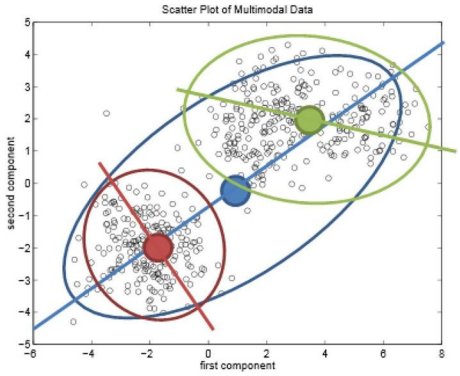
A. Overview of Applying GraphCut to Road Detection

With GraphCut, we view our road detection as a binary labeling problem as in [31] and [32]. An image I is formed by a set of pixels $I=\{v_{xy}: (x, y)\in\Omega\}$. For a color image, v_{xy} is viewed as an RGB vector, and for a gray image, it is an intensity value. We use (x, y) to denote the position of a pixel, and $\Omega\subset R^2$ is the image domain. Let $U=\{u_{xy},(x, y)\in\Omega\}$ be a set of labels, u_{xy} is assigned to the pixel at the position (x, y), which can be either 0 for nonroad or 1 for road defines the road areas where pixel labels are 1.

where ξ is the set of 8-neighboring pixel pairs. We $|\cdot|$ to denote the absolute value, allowing us to capture color difference only along the segmentation boundary. $\cdot$ is the L^2-norm factor. The parameter β is to control the smoothness and preciseness of the segmentation boundary, which is generally chosen to be as follows according to [31]:

A weighted graph is then constructed, where the cost on each graph edge is defined based on the terms E_c and E_s. The minimization of the Gibbs function becomes to find a cut with the minimum cost to partition the graph into two, which is solved by a min-cut/max-flow algorithm [32]. An iterative minimization scheme in [31], [33] can further improve accuracy.

Next, a detailed descriptions of the modeling of the color distributions using GMMs, image structure features capturing, and road detection using GraphCut is clarified.



Off-Line (Static) GMMs for Road/Nonroad

Color Modelling

Fig. 4. Illustration of the Orchard–Bouman splitting procedure on 2-D data: In the first iteration, the whole data set (blue oval) is split into two (green and red ovals) after projecting data onto the direction of the eigenvector (blue line) that corresponds to the largest eigenvalue of the covariance of all data. If the largest eigenvalue of the covariance of the data within the green oval is larger than that of the red oval, the data within the green oval is split into two groups next. The data within the red oval is splitted next. This process is repeated again and again as to achieve the desired number of clusters.

Road and nonroad pixels are first collected from sample images for learning road/nonroad color distributions. We select dozens of frames from UAV videos as the sample images, and scratch several strokes in each frame using green and red colors to specify road and nonroad pixels, respectively, as shown in Fig. 5(a). Two Gaussian mixture models GMM_0 with K_0 components and $GMM_1$with K_1components are used to represent the difference in road distributions. In this paper, we choose K_0 as 3 and K_1 as 5, which work well in our experiments.

The next action is performed for creating K_0 and K_1 components . This requires partitioning of nonroad pixels into K_0 clusters, and road pixels into K_1 ones. There are a large number of methods for data clustering [34]. In this paper, the applications of Orchard and

Volume 5 Issue III, March 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

Boumand binary splitting algorithm [35] is employed. It performs statistical analysis on data to fast pro-duce solutions, where in each iteration, the cluster with larger variance is split into two until the desired number of clusters is reached. As it is found Empirically, the Orchard and Boumand approach is twice faster than Kmeans in creating GMMs. Fig. 4 gives an illustration of the Orchard and Boumand splitting process on 2-data. Algorithm 1 clarrifies the procedure of GMM creation. Based on the clustering results, a GMM (for either road or nonroad) can be represented by using K triplets GMM = $\{(\mu_1, \Sigma_1, w_1), (\mu_2, \Sigma_2, w_2), \ldots, (\mu_K, \Sigma_K, w_K)\}$, where $\mu_i$and Σ_i are the mean color value and the 3 × 3 covariance matrix of a Gaussian component, respectively. The w_i denotes the weight of each Gaussian component. We set w_i as the number of pixels in the ith component over the total number of (road or nonroad) pixels

Algorithm 1 The creation of a GMM
Input: The number of components$K$and a set of road (ornonroad) pixels Γ.
Output:Kcomponents$C_1, \ldots, C_K$.
$C_m = \Gamma$, $m = 1$. $C_m$is a cluster with the largesteigenvalue.
For ($k = 2, \ldots, K$)
For C_m, calculate the mean value μ_m, the covariance matrix Σ_m, the largest eigenvalue λ_m and the corre sponding eigenvector v_m of Σ_m.
Split C_m into two sets, $C_k = \{c \in C_m : v_m c \leq v_m \mu_m\}$, and $C_m = C_m - C_k$.
For ($i = 1$upto k)
calculate the covariance matrix Σ_i, the largest eigen value λ_i of Σ_i.
If ($\lambda_i > \lambda_m$), $m = i$.
The calculation of (8) is to make st symmetric with respect to pixels at (x, y) and (i, j), where κ is a global parameter. In the definition of (9), the term $(i-x, j-y)S(x, y)(i-x, j-y)^T$is approximately $v_{xy} - v_{ij}^2$measuring the difference of color, and the other terms are used to give a pixel with large values of λ_+ and λ_+/λ_- a big weight of being on the segmentation boundary, referred to [36]. is a small number introduced to avoid the case that λ_+/λ_- is too big. $\lambda_+(x, y)$, and $\lambda_-(x, y)$ represents the maximal and minimal eigenvalues of the structure tensor S at (x, y), seeing (10). With these definitions, the more similar two colors are (or the smaller λ_+ and λ_+/λ_- are), the larger E_s, and thus less likely to be the segmentation boundary.
Fig. 5 illustrates E_c and E_s visually. The segmentation result is to assign the label "1" to the pixels with low P_{GMM0} and high P_{GMM1} values and the segmentation boundary goes along image structures. Morphology opera-tions, including erosion and dilation are performed to remove noises and fill holes. Contour analysis is applied to find large connected regions, which are the final road segmentation. we call it as the GraphCut detection approach based on static GMMs since the method relies on a static GMM model.
Structure Tensor: To easily interpret the structure tensor$S$, the color image I is viewed as a differentiable function $I(x, y) : \Omega \rightarrow R^3$. A 2 × 2 symmetric matrix defined in (10)with 3 × 3 window size is the structure tensor [36], which indicates local geometric structures of the image

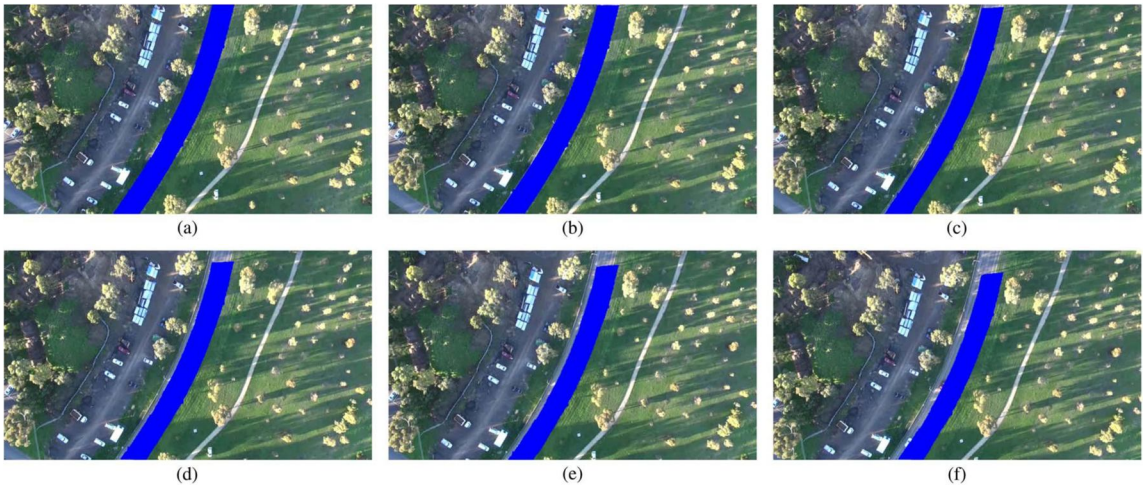


Fig. 6. View of the drift problem in homography-alignment-based road tracking. (a) Initialized road region in the first frame. (b) to

Volume 5 Issue III, March 2017
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

(f) Mapping the road region of (a) to the 20th, 40th, 60th, 80th, and 100th frames, respectively, by the transformation derived from the accumulative homographies. The drift problem in road tracking is problematic with more accumulations. To correct the drift error in tracking, we automatically locate the frame where it is necessary to switch from homography-alignment-based tracking to GraphCut detection for road area segmentation.

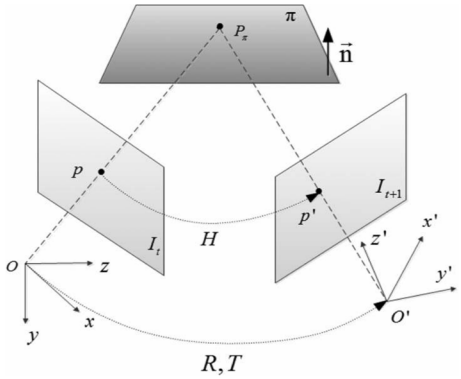
Image Downsampling: For an image with Npixels, theGraphCut algorithm has an average computational complexity of O(N logN), which is the bottleneck of our proposed system. To reduce the processing time, we perform road detection on a downsampled image, where the image resolution is reduced to a half by applying the bicubic interpolation algorithm.

V. HOMOGRAPHY-ALIGNMENT-BASED ROAD TRACKING

In this section, we give the details on how to achieve a fast road tracking based on homography alignment. We propose a rectification solution to the drift problem caused by accumulation error in homography estimation, and give criteria on assessing tracking results. Fig. 7 sources the pipelines of road tracking.

A. Fast Homography Estimation

We treat the ground scene as a plane π in the view of UAVs. Fig. 8 gives an illustration of the captured scene. For a 3-D point P_π , it has the coordinate p in I_t when we project P_π on the image I_t



Geometry homography transformation. in the camera coordinate system (O, x, y, z), and the coordinate p in $I_{t+1}$by projecting $P_\pi$on I_{t+1}in(O , x , y , z). The two camera coordinate systems are transformed by the rotation R and translation T . Suppose that the plane π is with a normal vector n. matrix H is explained by using R, T , and n is able to match the image points of the plane πbetween the two images I_t and I_{t+1} [37]. H is a 3 × 3 matrix with 8 freedoms: 3 for the rotation, 3 for the translation and 2 for the normal vector to the plane. Given $M \geq 4$ non-collinear corresponding point pairs (p_i, p_i) (i= 1, . . . M), H can be calculated. Using H, tracking of road regions in the current frame (i.e., I_{t+1}) by geometrically mapping the road regions in previous frame (i.e., I_t) to I_{t+1}. Fig. 1(a) and (b) illustrates the road region mapping from a previous frame to current one based on homography alignment.

To estimate H between two images, the most common way is firstly to detect sets of interest points in two frames, then find correspondence between the two sets of interest points through correlation-based matching, and finally estimate the homography matrix based on matched point pairs via a robust approach such as RANSAC [38]. Generally, the homography estimation in this way is computationally complex when the image resolution is as high as in our application, where the two computationally high parts are the point (or corner) matching and the RANSAC steps.

Therefore, To speed up the procedure of establishing correspondence between two sets of corners, we first apply the FAST approach [14] to detect interest points in each frame. Compared with some other well-known features such as SIFT [39], SURF [40], Harris [41], can achieve the best balance between accuracy and efficiency for our purpose. Next, the application of KLT tracker is employed to the set of detected FAST corners in each frame so that we can get the predicted FAST corners in the following frame. In other words, the KLT tracker gives correspondence between the detected FAST corners and the predicted ones in the following frame through optical flow technique. Note that the procedure of establishing corner correspondence. In the conventional way, the corners were detected in each frame and correlation is used to establish the matching between corners of two frames. It might work in most cases, where the two frames perhaps not close to each other, but the correlation is higher computational complexity. In contrast, our application is a special case, where the two frames to be aligned is consecutive, and the motions of corners are small, which can just satisfy the condition of the KLT tracker. Fig. 9(a) and (b) gives point pairs that are found by using tracking and

matching techniques, respectively. The similar points movements in Fig. 9(a) and (b) indicates the effectiveness of the proposed technique.

After finding the corresponding FAST corners of two consecutive frames through KLT, next, it is used to estimate the homography . . Since our purpose of using homography is to align the road region in two consecutive frames as accurately as possible, we propose a context-aware homography estimation approach, where only the corresponding FAST corners in the local neighborhood of road (a rectangular region which is obtained by expanding one quarter of the width of the road bounding box to left and right, respectively) are used within the RANSAC process. This is motivated by the observation that the homography that is estimated by RANSAC based on a local sampling of FAST-corner pairs can more accurately align the two images at this local region than the homography that is estimated by the common RANSAC . We illustrate the context-aware homography estimation scheme in Fig. 10 and show its advantage over the common RANSAC.

B. Road Tracking

The road area is tracked in a current frame by transforming the road region of a previous frame with the homography matrix *H*. Suppose the road contour in the previous frame is C_t. The road contoured track in the current frame takes the shape of $C_t^{H}{}_{+1}=HC_t$. The transformed contour achieves the road tracking in the current frame. Fig. 1(b) shows an example. On comparing previous frame, there are incoming regions with incoming road areas in a current frame. The incoming road area $R_t^{in}{}_{+1}$ need to be extracted and then combined

As UAVs provide fly forward facility, the incoming road appears near by the top of the aligned road. We thus define the ROI centered by the top point of the median axis of $C_t^{H}{}_{+1}$. By performing distance map calculation, skeleton, and noise removing we obtain median axes. After application of that, the algorithm in previously viewed Section IV is applied in ROI.

1) *Correcting Drift Error Occuring while Tracking:* As shown in Fig. 6, the drift errors tend to be very small in the first several dozen frames, and become larger with more incoming frames. The drift errors arise due to two factors: the first one is the inaccuracy in homography estimation. The second one is the ac-cumulation of round errors even when homography estimation is correct, because the transformation between two consecutive frames is slight and thus, the calculation is sensitive to round error. Large drift errors could arise with the accumulation of very small ones The large errors cause the aligned road area drifts far from the real road location. We propose a solution to deal with it. For the drift error caused by round error, considering that the drift errors are very small in the initial several dozen frames, the road regions in the first 25 frames are still tracked based on homography alignment between two consecutive frames. But for the subsequent frames, we do the road tracking by estimating the transformation between the current frame and the last tenth frame instead of the previous one, as shown in Figs. 6 and 17. The reason is that, RANSAC uses a small fixed threshold to control the inliers for the optimal transformation estimation. If the transformation between consecutive frames seems too small, RANSAC requires false inliers for transformation estimation. Because of this reason we estimate the transformation between two frames that are z frames apart. We can learn the "z" based on the median magnitude of the motion vectors shown in Fig. 9. For our aspect, z is set to 10.

In addition, to avoid large drift error, each tracking result is evaluated according to a criteria given in the following, so that if the tracking is unsuccessful, online GraphCut detection is applied instead. Specifically, we propose to compare the corresponding RGB histogram of the current tracking result with an average RGB histogram obtained from the tracking results of the last 10 frames. If the difference is noted as above a threshold, the tracking result drifts far away from the real road area. The histograms can built with bins for each color channel. In each bin, the pixel number percentage in the bin over the total number of pixels is estimated . Moreover, large accumulative error in tracking may also cause obviously zigzag contours, as shown in Fig. 18. We check each road contour. If the change of contour length is above a threshold, tracking results are treated as unsuccessful.

VI. EXPERIMENTAL RESULTS

Experiments are mainly conducted on image sequences ac-quired using our UAV that flew in different sessions near our center in Australia. Image sequences downloaded online are also used for evaluation on different scenarios. In the test data set, there are 2780 images from six videos of different resolutions, of type including 1280 × 720, 1244 × 748, 1024 × 576, and 848 × 480. Owing to limitation of used UAV flight, we are abide of capture videos of unpaved roads. Paved Roads are utilized in our experiments . Inappropriate conditions such as slow/fast UAV movements, low/high-altitude flying, existence of a lot of shadows, and large variations on image scenes are included in the data set. The colors and shapes of road are different from each other.

In this paper, quantitative evaluations are provided by com-paring the results with ground truth pixel wisely using two

measurements: 1) precision $Q=TP/(TP+FP)$, which is the percentage of correctly classified road pixels over the total detected road pixels; and 2) error rate $ER=(FP+FN)/(TP+FN)$, which is the percentage of wrong classified image pixels over the ground-truth road area. TP, FP, and FN, respectively, are true positive, false positive, and false negative. is combined with the online GraphCut detection in ROI as the tracking result and then the tracking result is pixel wisely com-pared with the ground truth. From the result in Table II, we can see that our homography-alignment-based technique is much faster and more accurate. The reason is the introduction of the KLT tracker and context-aware RANSAC. The further experiment on the context-aware homography estimation scheme is given in Figs. 10 and 11. We can see that the alignment of road regions based on our context-aware homography-alignment-based tracking is more accurate than the one obtained by the traditional homography estimation. The reason can be due to the fact that some objects such as bushes, tress, or highland (hills) play a nontrivial role in homography estimation through the common RANSAC. However, these high objects cannot be treated as lying in the same plane with road areas, and are not desirable for the estimation of an optimal homography to accurately align road regions only. In addition, our context-aware homography estimation scheme is much faster than the common RANSAC one, and the computational time is less than 10%.

We also experiment different features for homography estimation, including SIFT, Harris, and SURF. Amongst them, SURF has been widely used for an accurate homography estimation. It takes moreover 0.142 s per frame for the detection of SURF features. FAST has the best overall performance (as studied in [14]), where the FAST feature detection only takes 0.008 s per frame.

Fig. 17 shows homography-alignment road tracking on frames with 10 intervals in-between, in comparison with Fig. 6. With the accepted level of interval, homography estimation becomes more speedest to calculation (round) error. Thus, road tracking can be successfully runned on more frames with zero drift-error correction. For the continuous frame-based estimation, road tracking often jumps after 10–30 frames, and for the interval-based estimation, road tracking fails after 100–300 frames.

A. Road Detection and Tracking

The proposed technique of road detection and tracking is experimented as a whole in this section. The technique is tested on UAV videos acquired at different altitudes and fly speeds (For our UAV, we fly with absolute attitudes of 130–170 m and ground speeds of 5–10 m/s). More sequences can be found from the supplementary file. Accurate estimates of road areas are provided in our experimental results. The switch between road detection and tracking only happens when there are too much changes of color distributions or/and contour lengths of tracked roads, and it can successfully solve the drift and zigzag problems.

Table III lists the running time of previous approaches on road detection and tracking. The work done by Frew *et al.* [4] is implemented in this concept using static GMM on original video frames with the 848 × 480 resolution. The other statistics are obtained from the papers. Our proposed approach challenges fastest road detection and tracking. We test 2760 frames in total with the resolution of 1046 × 595 on average. Tit is found that the mean detection time duration is 0.283s and tracking time is 0.028 s per frame. Road detection is required in 16 frames, including the initial detection and switching from unsuccessful tracking. The average running time of presented approach is 0.029 s per frame

VII. CONCLUSION

In this paper, a novel approach for road detection and tracking in UAV videos has been proposed. We utilize the static GraphCut then track road areas in subsequent frames by combining a fast context-aware homography-alignment road tracker and an online GraphCut approach for road detection in incoming ROIs. Efficiency and effectiveness of the proposed tracking technique are demonstrated in our experiments.

Based on our experiments, we notice that drift error and zigzag contour problems more often happen in UAVs at low altitudes and in high speeds. The reason is that (1) there are more high objects that are not in the same road plane when UAVs flying at low altitude Results with high Q and low ER are desired. The ground truth road areas are manually specified on 400 images evenly sampled from our test videos. All the experiments are conducted using C++ implementation on Intel i7-2600K 3.

There are some parameters introduced in our method. The values lotted in our experiments are listed as follows. The λ in (1) is to be 50 as in [36]. The κ in (8) is chosen to be $EP(s(v_{xy}, v_{ij}) + s(v_{ij}, v_{xy}))$. The $K_0$and $K_1$are set to be 5 and 3 for non-road and road GMMs. The size occupied by downsampled version in GraphCut detection is the half of parental images. The interval corresponding of updating GMM is 40 frames. The size of the incoming ROI is $w \times h$ with h= 60 and w= 6d, where d is the average distance of points on the median axis to the contour image downsampled to a half resolution and then the final result is obtained by scaling the intermediate result to the full-size images.

The first experiment is to evaluate GMM modeling, structure tensor, and GraphCut. We compare their performances to road

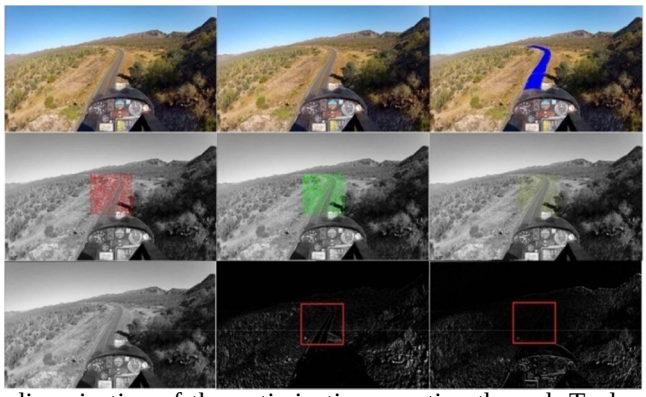
detection in terms of average error rates *ER*, precision *Q* and times *T* . The visual comparisons are given in Fig. 12. Note that the term of GMM based detection in Table I comes from [4] that detects road based on the probability map of (6). Since our road tracking is highly dependent on the detection result, in this stage the accuracy of detection is more important than efficiency.

The second experiment is fully relies on GMM components. Different numbers for K_0 and K_1 are tested in road detection. Mathematical details of average detection error rates and timings (i.e., *ER* and *T*) is shown for comparison. The result indicates that K_0= 5 and K_1= 3 achieve the best balance between efficiency and accuracy.

Fig. 14 compares the performances of the static GraphCut and online GraphCut in road detection. It can be seen that due to flight, colors of road, and background scene change a lot. As the dynamic GMM adaptively updates the model using new road and non-road pixels, it ensures more robust performance. Road detection on the original and half-sized is evaluated in Fig. 15. It is observed schematically that reduction of image size does not change much accuracy of road detection and does speed-up road detection.

A. Road Tracking

Road tracking is evaluated on video sequences when the detected roads are given. The first experiment is on the homography-alignment-based tracking schemes. We compare our with the traditional mechanism using FAST detection in images, correlation-based matching and homography estimation based on all matched point pairs. The corresponding times *T*, precisions *Q* and average error rates *ER* are compared in Table II, as well as the time for each step.



the condition of KLT tracker (the linearization of the optimization equation through Taylor expansion at very local region) is not satisfied when UAVs fly very fast. The proposed context-ware homography estimation solves the problem from (1), but it is still difficult for urban areas with high buildings along road sides. Switching from road tracking to detection can address the problem from (2), but it would be much slower since detection is more computationally complex than tracking. More intelligent methods similar to key frame-based [42] or incremental structure-from-motion [43] can be investigated

B. Performance Comparison of Road Detection and Tracking Techniques

Methods	Running time	Image resolution
Wang *et al* [12]	4s	240 × 256
Kim *et al* [25]	0.06s	352 × 240
Guo *et al* [29]	0.125s	320 × 240
Alvarez *et al* [28]	0.6s	640 × 480
Siogkas *et al* [7]	0.35s	640 × 480
Frew *et al* [4]	0.35s	848 × 480
Our approach	0.029s	1046 × 595

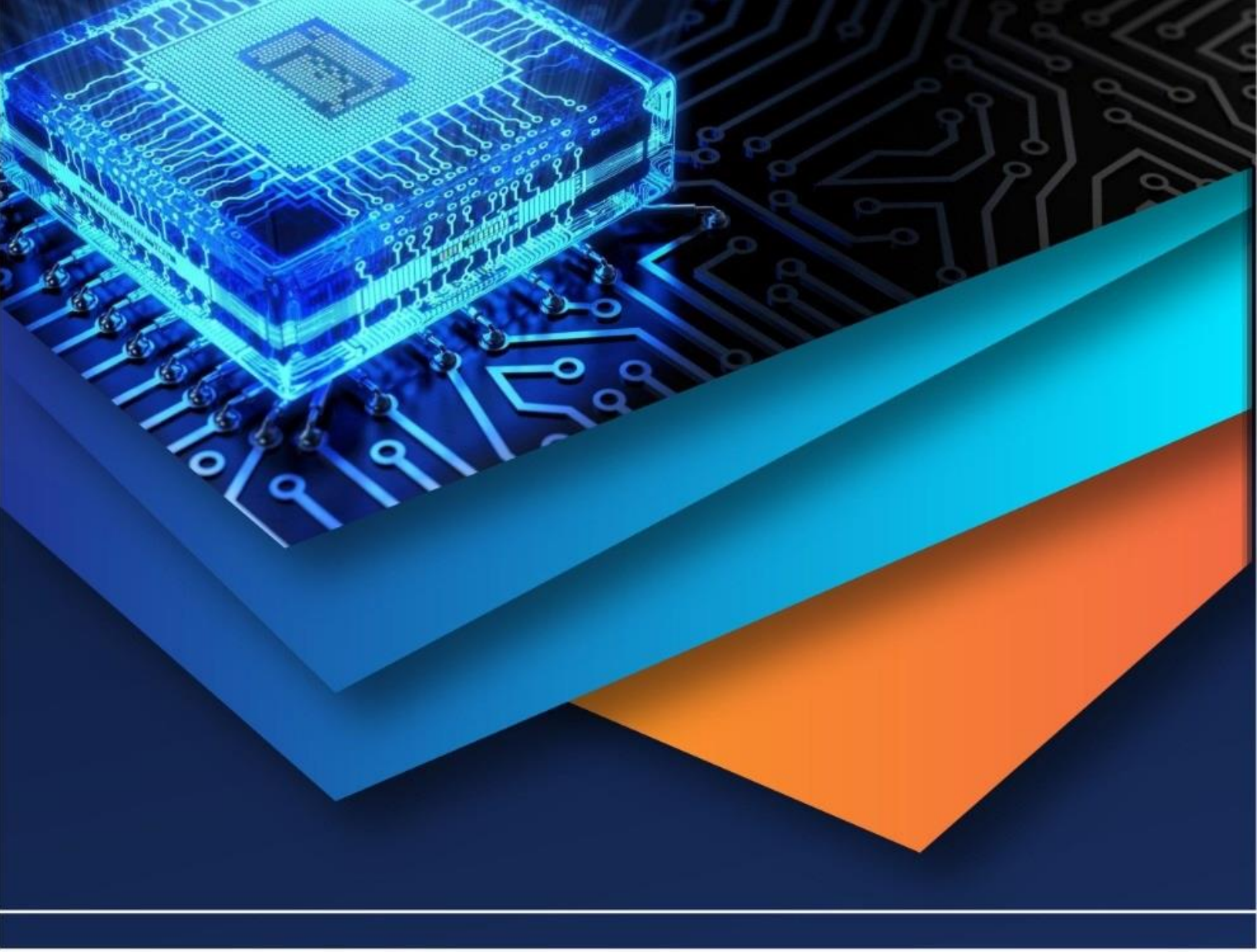
VIII. ACKNOWLEDGMENT

The authors would like to thank M. Fielding for flying UAV.

REFERENCES

[1] Coifman, M. McCord, M. Mishalani, and K. Redmill, “Surface trans-portation surveillance from unmanned aerial vehicles,” in *Proc. 83rdAnnu. Meet. Transp. Res. Board*, 2004, pp. 1–9.
[2] Online]. Available: http://www.list.ufl.edu/ATSS.ht
[3] [Online]. Available: http://soliton.ae.gatech.edu/people/ejohnson/

[4] E. Frew *et al.*, "Vision-based road-following using a small autonomous aircraft," in *Proc. IEEE Aerosp. Conf.*, Mar. 2004, vol. 5, pp. 3006–3015.
[5] H. Zhou, D. Creighton, L. Wei, D. Y. Gao, and S. Nahavandi, "Video driven traffic modeling," in *Proc. IEEE/ASME Int. Conf. Adv. Intell.Mechatronics*, Jul. 2013, pp. 506–511.
[6] Y. He, H. Wang, and B. Zhang, "Color-based road detection in urban traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 5, no. 4, pp. 309– 318, Dec. 2004.
[7] G. K. Siogkas and E. S. Dermatas, "Random-walker monocular road detection in adverse conditions using automated spatiotemporal seed se-lection," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 2, pp. 527–538, Jun. 2013.
[8] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimiza-tion via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.
[9] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," in *Proc. IEEE Comput. Vis. Pattern Recog.*, 2000, pp. 142–149.
[10] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image Vis. Comput.*, vol. 21, no. 1, pp. 99–110, Jan. 2003.
[11] B. K. P. and B. G. Schunck, "Determining optical flow," *Artif. Intell.*, vol. 17, no. 1, pp. 185–203, Aug. 1981
[12] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-Snake," *Image Vis. Comput.*, vol. 22, no. 4, pp. 269–280, Apr. 2004
[13] Y. Wang, D. Shen, and E. K. Teoh, "Lane detection using spline model," *Pattern Recog. Lett.*, vol. 21, no. 8, pp. 677–689, Jul. 2000
[14] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 430–443.
[15] C. Tomasi and T. Kanade, "Detection and tracking of point features," in *School of Computer Science*. Pittsburgh, PA, USA: Carnegie Mellon Univ., 1991
[16] S. Rathinam, Z. Kim, and R. Sengupta, "Vision-based monitoring of locally linear structures using an unmanned aerial vehicle 1," *J. Infras-tructure Syst.*, vol. 14, no. 1, pp. 52–63, Mar. 2008.
[17] S. Rathinam *et al.*, "Autonomous searching and tracking of a river using an UAV," in *Proc. IEEE Am. Control Conf.*, Jul. 2007, pp. 359–364.
[18] Y. Lin and S. Saripalli, "Road detection from aerial imagery," in *Proc.IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3588–3593.
[19] Y. Lin and S. Saripalli, "Road detection and tracking from aerial desert imagery," *J. Intell. Robot. Syst.*, vol. 65, no. 1–4, pp. 345–359, Jan. 2012.
[20] A. Angel, M. Hickman, P. Mirchandani, and D. Chandnani, "Methods of analyzing traffic imagery collected from aerial platforms," *IEEE Trans.Intell. Transp. Syst.*, vol. 4, no. 2, pp. 99–107, Jun. 2003.
[21] B. Coifman, M. McCord, R. G. Mishalani, M. Iswalt, and Y. Ji, "Roadway traffic monitoring from an unmanned aerial vehicle," *Proc. Inst. Elect.Eng.- Trans. Intell. Transp. Syst.*, vol. 153, no. 1, pp. 11–20, Mar. 2006.
[22] A. P. DalPoz, R. A. B. Gallis, J. F. C. daSilva, and E. F. O. Martins, "Object-space road extraction in rural aeras using stereoscopic aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 4, pp. 654–658, Jul. 2
[23] J. Hu, A. Razdan, J. C. Femiani, M. Cui, and P. Wonka, "Road network extraction and intersection detection from aerial images by tracking road footprints," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4144– 4157, Dec. 20
[24] C. Unsalan and B. Sirmacek, "Road network detection using probabilis-tic and graph theoretical methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4441–4453, Nov. 2012.
[25] Z. W. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 16–26, Mar. 2008
[26] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, vol. 2, pp. 96–103
[27] H. Kong, J. Y. Audibert, and J. Ponce, "General road detection from a single image," *IEEE Trans. Image Process.*, vol. 19, no. 8, pp. 2211–2220, Aug. 2010.
[28] J. M. Alvarez and A. M. Lopez, "Road detection based on illuminant invariance," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 1, pp. 184–191, Mar. 2011.
[29] C. Guo, S. Mita, and D. McAllester, "Robust road detection and tracking in challenging scenarios based on Markov random fields with unsuper-vised learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1338– 1354, Sep. 2012.
[30] A. Xu and G. Dudek, "A vision-based boundary following framework for aerial vehicles," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Oct. 2010,b81–86.
[31] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut-interactive foreground extraction using iterated GraphCuts," *ACM Trans. Graph.*, vol. 23, no. 3, 309–314, Aug. 2004.
[32]
[33] Y. Boykov and M. Jolly, "Interactive graphcuts for optimal boundary and region segmentation of objects in n-d images," in *Proc. Int. Conf. Comput.Vis.*, 2001, pp. 105–112.J. F. Talbot, Implementing Grabcut. [Online]. Available: http://www. justintalbot.com/
[34] S. J. Sangwine and R. E. N. Horne, *The Color Image Processing Hand-book.* New York, NY, USA: Springer-Verlag, 1998.
[35] M. T. Orchard and C. A. Bouman, "Color quantization of images," *IEEETrans. Signal Process.*, vol. 39, no. 12, pp. 2677–2690, Dec. 1991
[36] H. Zhou, J. Zheng, and L. Wei, "Texture aware image segmentation using GraphCuts and active contours," *Pattern Recog.*, vol. 46, no. 6, pp. 1719– 1733, Jun. 2013
[37] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vi-sion*, vol. 2. Cambridge, U.K.: Cambridge Univ. Press, 2000
[38] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated car-tography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981
[39] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008
[40] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, vol. 2, pp. 1150–1157
[41] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–151.
[42] A. Rahimi, L. Morency, and T. Darrell, "Reducing drift in differential tracking," *Comput. Vis. Image Understanding*, vol. 109, no. 2, pp. 97– 111, Feb. 2008.
[43] C. Wu, "Towards Linear-Time Incremental Structure from Motion," in *Proc. IEEE Int. Conf. 3-D Vis.*, Jun./Jul. 2013, pp. 127–134



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY