

# Plantation Monitoring and Yield Estimation using Autonomous Quadcopter for Precision Agriculture

Vishakh Duggal<sup>1</sup>, Mohak Sukhwani<sup>1</sup>, Kumar Bipin<sup>2</sup>, G. Syamasundar Reddy<sup>1</sup>, K. Madhava Krishna<sup>1</sup>

**Abstract**—Recently, quadcopters with their advance sensors and imaging capabilities have become an imperative part of the precision agriculture. In this work, we have described a framework which performs plantation monitoring and yield estimation using the supervised learning approach, while autonomously navigating through an inter-row path of the plantation. The proposed navigation framework assists the quadcopter to follow a sequence of collision-free GPS way points and has been integrated with ROS (Robot Operating System). The trajectory planning and control module of the navigation framework employ convex programming techniques to generate minimum time trajectory between way-points and produces appropriate control inputs for the quadcopter. A new ‘pomegranate dataset’ comprising of plantation surveillance video and annotated frames capturing the varied stages of pomegranate growth along with the navigation framework are being delivered as a part of this work.

## I. INTRODUCTION

The current farming practices for yield estimation are labour intensive and expensive. They scale poorly for large commercial farms and are not consistent. Whereas, precision agriculture utilizes real time processing of a site-specific farming information to minimize the input cost, improve efficiency and is scalable for large farms. Satellite images [1] have played a major role in such real time deployments and are being massively used in the present day agricultural infrastructure. However, in recent times we have witnessed a surge in use of quadcopters in agricultural assistance. They have overcome some of the traditional challenges of satellite imagery – low resolution, infrequent revisit times and distortions due to cloud cover. Quadcopters are seen as one of the potential platforms for precision agriculture in near future.

The solution presented in this work is applicable to diverse plantations, with its capabilities being demonstrated over organic pomegranate plantation. The capabilities of the proposed approach could be described as a twofold framework (1) plantation monitoring and yield estimation: life cycle stage grading and yield estimation of pomegranate while autonomously navigating through inter-row paths of the plantation using a monocular camera as its primary sensor. (2) Autonomous navigation framework: a novel navigation and control framework, which uses GPS way-points as input provided by a farmer and employs convex optimization for generating minimum time trajectory and control.

First, the plants in the farms are mostly planted systematically in row arrangement resulting in parallel paths

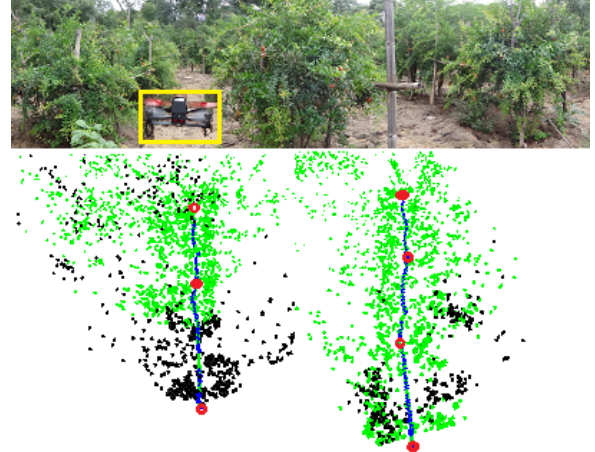


Fig. 1: Top Image: Various rows of the pomegranate plantation and the quadcopter flight (yellow box). Bottom image: Trajectory traversed by the quadcopter in two rows of the pomegranate plantation (Blue line). Red circles correspond to GPS ways points whereas green and black dots are used to mark reference points computed by ORB-SLAM.

of uniform width between the plants rows (inter-row) to facilitate the movement of workers around the farm, Fig. 1. Pomegranates have two major life cycle stages – flower bud and ripened fruits. The curation of later provides current yield estimates and the former correspond to future yield estimates. Current practices for yield estimation focus mainly on ripened fruits population estimation [2]. Earlier methods for yield estimation were either restricted to laboratory [3] or used color [4] for estimating yield. Cihan et. al [5] presented pomegranate yield estimation using segmentation based on the color and shape features. Their method falls short in scaling to cases that involves variations in lighting and occlusion. However, Calvin et. al [2] did fruit segmentation using special RGB-IR camera setup, making them expensive and hard for real time deployments.

One of the major contributions of our work, it specifically addresses these issues by using color and texture cues [6] as feature vector calculated over each of multiple generated candidate objects [7] and employ supervised learning framework using SVM (Support Vector Machine) for detection. Moreover, quadcopter flies autonomously through the inter-row path using GPS way-points at the average height of pomegranate plants while estimating the yield of pomegranates and flower buds using a monocular camera.

Second, precise navigation and control is the core of all automation. We developed a novel navigation framework for Bebop quadcopter by Parrot Corporation [8]. This quadcopter is specifically selected due to its low cost, high stability,

<sup>1</sup>The authors are with IIIT Hyderabad, India. { vishakh.duggal, mohak.sukhwani }@research.iiit.ac.in, { mkrishna, shyam.reddy }@iiit.ac.in

<sup>2</sup>Associated with Uurmi Systems Lab, Hyderabad, kumarb@uurmi.com.

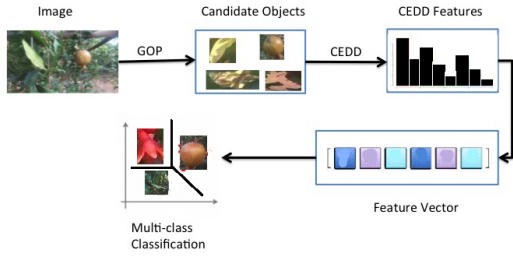


Fig. 2: Process flow of pomegranate and flower bud detection: The input image is segmented into various candidates boxes. CEDD features computed over these candidate selections are subjected to a multi-class classifier which segregates each the candidates into flower, pomegranate or NA.

inbuilt GPS and ability to adjust the angle of the high resolution camera independent of the quadcopter motion [8]. The framework is able to navigate quadcopter autonomously using collision free GPS way-points provided over the plantation by a farmer. Moreover, it is capable of handling the unplanned drift and maintain heading along the middle of an inter-row path.

Another notable contribution of our work is the development of the autonomous navigation framework for Bebop, which uses convex optimization for generating a minimum time trajectory between way-points over Robot Operating System (ROS) middle-ware, “*first to release in Open Source*”. This we believe could facilitate further research by providing standard interface and documentation for efficient maintenance and integration with the third party modules. To the best of our knowledge, such a solution for plantation monitoring and yield estimation with autonomous navigation framework using commercial quadcopter has not been presented in literature before.

We begin by describing the utility of quadcopters in precision agriculture. Monitoring and yield estimation of pomegranate plantation and various associated challenges are discussed in Section II-A. In Section III, we discuss details of the proposed autonomous navigation framework for the quadcopter. The experiments and results described in Section IV demonstrate the robustness and utility of our solution.

## II. QUADCOPTER IN PRECISION AGRICULTURE

This section describes an overview of the proposed framework which performs plantation yield forecasting using generic monocular quadcopter. The proposed framework has been evaluated on a low cost commercial quadcopter Bebop [8] equipped with frontal 14 Mega pixel “fish-eye” camera and records video in an  $180^\circ$  field of view with  $1920 \times 1080$  resolution in its on-board memory. We specifically use this for yield estimation of pomegranates and its flower buds. At the same time it even transmits the video stream of  $640 \times 468$  resolution over Wifi link which is used by navigation module for the perception of environment and sate-estimation.

### A. Plantation Monitoring and Yield Estimation

The proposed framework follows three main stages for yield prediction – segmentation, feature extraction and classification. Existing fruit segmentation algorithms [9]–[12]



Fig. 3: Variations in selected candidate objects: The candidate bounding boxes extracted from images are of varied shapes and sizes. They don’t follow a standard shape pattern making the classification task challenging.

work either on color or shape, which is susceptible to both lighting and occlusion. Moreover, object recognition algorithms either follow the sliding window [13] approach or use segmentation techniques to extract multiple candidate objects which are analysed later. Our proposed framework is motivated by research of Philipp et al. [7] which uses automatic segmentation to generate multiple candidate objects  $N$ , shown in Fig. 3. The state of art method initially, for an image, computes superpixels and a boundary probability map that associates a boundary probability with each superpixel edge. Set of seed superpixels are calculated amongst them and based on these foreground and background masks are calculated. Afterwards, signed geodesic distance transform is calculated over the masks with critical set among them becoming the candidate objects [7], later used for recognition in our paper. These candidate objects are provided as input to the trained system and classified as either fruit, flower bud or others with respective label  $L$  and confidence index  $CI$  as output.

To make the detection resilient to lighting, occlusion and direction our framework extracts monocular cues, viz. color and texture features, from each candidate object and stores in the form of the histogram- Color and Edge Directivity Descriptor (CEDD) [6]. The size of the histogram is 144 bins with each bin represented by 3 bits ( $144 \times 3 = 432 \text{ bits}$ ). The histogram is divided into 6 regions, each determined by the extracted texture information. Each region is further divided into 24 individual sub regions each comprising color information  $6 \times 24 = 144$ . The HSV color channel is provided as input to a fuzzy system to obtain color information. The texture information is composed of edges characterized as vertical, horizontal,  $45^\circ$ ,  $135^\circ$  and non-directional [6]. This histogram is used as a feature vector to describe the candidate object and provided as input to the learning framework.

Due to the unavailability of the standard training set,



Fig. 4: Detected pomegranates and flowers (in Green boxes). False negative are marked (with red box) and False positives (with blue box) (manually for visualization)

comprehensive training and testing set are created using real quadcopter monitoring video. The framework uses a semi-supervised method to label the dataset by annotating every fourth frame and then propagating the results to remaining frames. In all 5000 frames are manually annotated and subsequently the annotations are propagated on rest of the 15000 frames. Adjustments to the annotations (if any) are done by visiting and verifying annotations of each frame. To increase the robustness of the training data towards occlusion, for every positive annotation additional annotated boxes are generated by repositioning them by  $3\text{pixels}$  across 8 directions around original bounding boxes. In this way partially occluded fruits and flower buds dataset is created. Overall, 25400 labeled ‘pomegranate’, 5000 labeled ‘buds’ and 54000 ‘NA’ are there in our dataset, we use 4:1 split for training and testing purpose in experiments.

Plantation monitoring and yield estimation is conceptualized as a multi-class classification problem. The learning framework determines label  $L_i = \{\text{pomegranate, flower bud or 'NA' (leaves, sand, trunk etc.)}\}$  of candidate object with confidence index  $CI_i$ ,  $\forall i = 1 \dots N$  of the respective label. The system is trained using libSVM package with Radial Basic Function (RBF) kernel model capturing the non-linear characteristics of the training dataset, Fig.2. Training is done in an iterative manner, starting with small training dataset, part of false positives and negatives of previous training iterations are appended to training dataset for subsequent cycles till training accuracy stagnates. This results in improved accuracy and faster convergence with the system capable of handling diverse environments. The model files thus generated are used for classification score computation.

For a given frame of an input video, the multiple candidate objects identified by [7] are used as an input to the supervised learned system. These sets of candidates are thereafter classified into pomegranate, flower bud or others labels  $L_i$ ,  $\forall i = 1, \dots, N$ . Candidate objects which are labeled either as fruit or flower bud are represented by a bounding box  $\{Pos_j, Ar_j, L_j, CI_j\}$ ,  $\forall j = 1 \dots K$  and  $K \leq N$ , where  $K$  number of objects classified as pomegranate or flower bud,  $Pos_i$  is top-left corner position bounding box for candidate object,  $Ar_j$  is the area of the box,  $L_j$  label of object and  $CI_j$  confidence index of label. Candidate objects below a threshold of confidence index are rejected to reduce false positives. Non-maxima suppression, NMS is applied on overlapping regions with similar labels and only highest scoring boxes are retained Fig. 4.

### III. AUTONOMOUS NAVIGATION FRAMEWORK

Autonomous navigation is a paramount requirement for precision agriculture using quadcopters. This section discusses the novel navigation framework developed for autonomous navigation of Bebop quadcopter in an outdoor environment. Bebop quadcopter is a low cost, self stabilizing, with inbuilt Wifi Hotspot, GPS, IMU and ability to adjust the orientation of the high resolution monocular camera digitally, independent from the motion of the quadcopter. It also provides efficient communication for control commands and

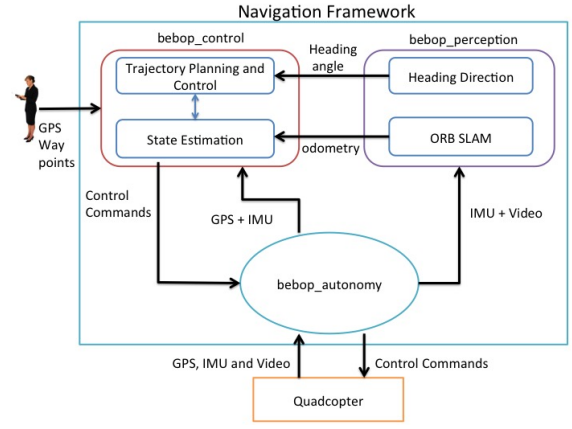


Fig. 5: Navigation framework: Each block represents individual component of the navigation framework. Information flow across the components is represented by arrows between the blocks.

data (video and IMU) over 2.4 Ghz Wifi link with host device [8]. It provides two video streams at different resolution captured from the same monocular camera, high resolution  $1920 \times 1080$  video is directly saved in on-board memory of the quadcopter and is primarily used for monitoring and yield estimation of pomegranate plantation. Lower resolution  $640 \times 438$  video is streamed real time over Wifi connection to host device and used primarily for navigation and state estimation. The navigation framework is developed over ROS middle-ware and released in *Open Source* to allow efficient maintenance and integration with the third party modules. The framework development involved software interface for communication link to quadcopter, state estimation, perception, and trajectory planning and control modules over ROS middle-ware.

The proposed framework for ROS has three major packages—*bebop\_autonomy*, *bebop\_control* and *bebop\_perception*, Fig. 5. *bebop\_autonomy* handles the execution of the low level control commands, wifi networking, transfer of IMU data and transfer of the decoded video stream from quadcopter to the host device in real time. *bebop\_control* is the most comprehensive package which provides an interface for the teleoperation for manual control in emergency conditions, state estimation using IMU, GPS and ORB SLAM [14] along with trajectory planning and control for autonomous navigation. *bebop\_perception* manages local way-point generation and acts like an interface for ORB SLAM. More details are available on ROS wiki: <http://tinyurl.com/oc4uwlm>.

#### A. State Estimation

Accurate state estimation is imperative for autonomous navigation. Especially in the case of aerial vehicles where absence of odometry data from moving wheels makes the state estimation extremely challenging. This problem is addressed either at the global level using GPS, or local level using GPS and IMU [15] or visual SLAM [16]. The proposed framework utilizes fusion of data from IMU and visual SLAM with extended Kalman filter (EKF) to address the



limitations of systems using only GPS or IMU, Fig. 7. Initial monocular visual SLAM implementation such as PTAM [16] had limitation such as instability and breakage due to sudden camera motion, human intervention required for map initialization and restricted to small scenes and distances. These limitation have been successfully overcome with ORB SLAM implementation by Rau'l et.al [14] and is used in our application for state estimation. ORB-SLAM utilizes fast ORB features instead of SIFT-SURF for tracking, mapping and is more robust to breakage due to the camera movement. It is also highly stable and scalable across the outdoor environment. Monocular visual SLAM including ORB-SLAM provides relative odometry position data  $\{X_{map}, Y_{map}, Z_{map}\}$  in its map coordinate frame:

$$\{X_w, Y_w, Z_w\} = \alpha * \{X_{map}, Y_{map}, Z_{map}\} \quad (1)$$

where  $\{X_w, Y_w, Z_w\}$  are real world local coordinates and  $\alpha$  being respective scaling factor. Determining the scaling factor requires manual human intervention in PTAM, therefore an automatic algorithm was developed and integrated with ORB-SLAM by using IMU data from the quadcopter. At the time of the initialization of ORB-SLAM, quadcopter hovers at height  $h_{low}$  and later increases its height to  $h_{high}$  before reducing height again to  $h_{low}$ . Real world heights obtained from IMU  $Z_{w(low)}$  and  $Z_{w(high)}$  are recorded at  $(h_{low}, h_{high})$  along with relative  $Z_{map(low)}$  and  $Z_{map(high)}$  generated by the ORB-SLAM in its map coordinate frame to calculate scaling factor:

$$\alpha = \frac{(Z_{w(high)} - Z_{w(low)})}{(Z_{map(high)} - Z_{map(low)})} \quad (2)$$

Scaled odometry data obtained from ORB-SLAM which corresponds to the real world local coordinates is fused with IMU data. Altitude from ultrasonic sensor and velocity obtained using optical flow from secondary downward facing camera using Extended Kalman Filter (EKF) and later used with GPS way-points to obtain more refined state estimation.

Globally GPS navigation is the main constituent of any computer aided navigation either in the air, over land or sea. Quadcopter used in the experiment has an inbuilt GPS module and provides real time latitude and longitude data in the degrees form representation. Both longitude and latitude information needs to be converted into a global coordinate system before using it for navigation. There are multiple global coordinate systems available East-North-Up (ENU) coordinate system, Earth-Centered-Earth-Fixed (ECEF) coordinate system and most commonly used Universal Transverse Mercator (UTM) Cartesian coordinate system for navigation. The UTM cartesian coordinate system is used in our proposed framework as it is a horizontal position representation, i.e. identify locations on the Earth independent of vertical position and require simple euclidean metric calculations to determine the distance between any two points. The UTM system divides the Earth into sixty zones, each being a six-degree band of longitude with each zone following East-North frame representation. The UTM coordinates are then converted to local navigation coordinates by rotating each

co-ordinate value with  $(\delta)$ -yaw of the quadcopter, which corresponds to angle with magnetic north.

$$\begin{bmatrix} X_{local} \\ Y_{local} \\ Z_{local} \end{bmatrix} = \begin{bmatrix} \cos \delta & \sin \delta & 0 \\ -\sin \delta & \cos \delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} E \\ N \\ Z_w \end{bmatrix} \quad (3)$$

Where (E, N) represents east, north in the horizontal frame of the UTM and  $(X_{local}, Y_{local}, Z_{local})$  corresponds to a GPS way-point in local navigation coordinate frame. These are periodically fused with IMU and ORB-SLAM generated odometry using Extended Kalman Filter (EKF) to obtain refined odometry and applied in the trajectory planning and control of the quadcopter.

### B. Way-Point Navigation

Based on the area of interest in the plantation, the farmer provides GPS way-points in an ordered manner of visit- first point corresponds to start location and last to destination. In between the intermediate way-points define the path of the quadcopter. The navigation framework autonomously navigates quadcopter from the current position to the next way-point using trajectory planning and control module which uses convex optimization to generate minimum time trajectory, which is based on our previous work [17].

Monitoring and yield estimation of pomegranate plantation requires the quadcopter to navigate autonomously at an average height of pomegranate plants, through the inter-row path. Due to unplanned drift in motion or error in state estimation prevalent in quadcopters, the drone may lean and crash into plantation rows. To avoid such a situation, a visual feed back mechanism was developed to align the drone to the middle of the inter-row path while moving towards destination GPS way-point. Quadcopter monocular camera video feed of  $640 \times 468$  is processed and the inter-row path is segmented from its surroundings using Expectation-Maximization (EM) based unsupervised learning algorithm.

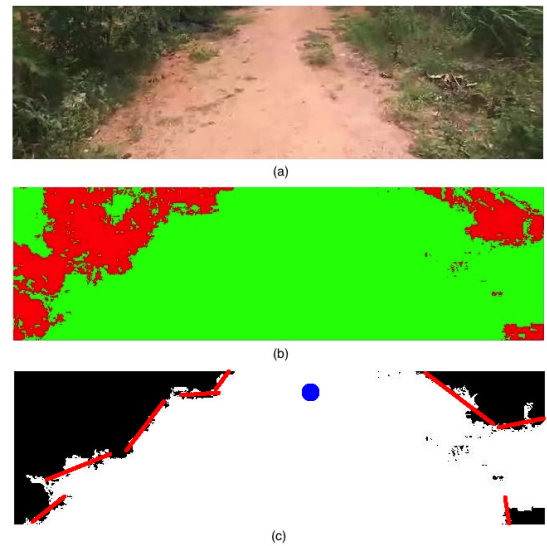


Fig. 6: Path Segmentation: (a) Image captured from quadcopter. (b) Segmented path from image with unwanted segmented objects (c) filtered image using morphology transformation with path clearly segmented from surroundings and vanishing point after kalman filter visible (blue dot) shown in figure for visibility.

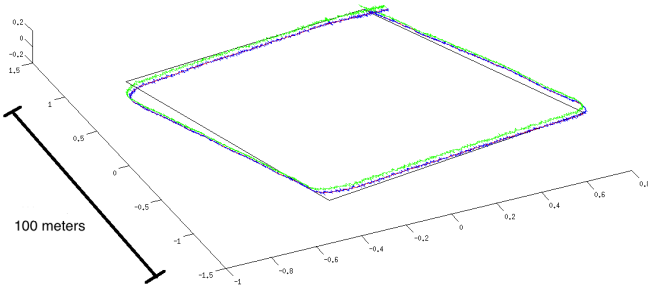


Fig. 7: State estimation of the quadcopter using ORB Slam and IMU with (Black line) ground truth, (Blue line) ORB slam visual odometry, (Green line) IMU state estimation and (Red Line) ORB slam and IMU fused Odometry

It estimates the parameters of the multivariate probability density function in the form of Gaussian mixture models. The segmented image contains inaccurately segmented small surrounding objects which are filtered by applying Morphology transformations, shown in Fig. 6. The proposed framework exploits the proven concept of vanishing point to maintain quadcopter along the middle of the inter-row path while navigating towards next GPS way-point. Initial lines are detected in the segmented path using LSD-Line Segment Detector [18]. The path segmentation improves both the accuracy and speed of the lines detections. The LSD algorithm when applied directly (without path segmentation) over an image, has far too many noisy lines detections when compared to LSD on the segmented path. The lines should converge towards a point but because of noise, all the lines do not intersect at exactly one point. The region in the image which has the highest density of pair-wise line intersections, indicates a high confidence index and contains the vanishing point. To achieve this, the image plane is divided into a  $M \times M$  grid  $G$  and middle of grid element  $G_{p,q}$  with maximum intersections is selected as the vanishing point.

$$(p, q) = \arg \max_{(p, q)} G_{p, q} \quad (4)$$

The confidence index and presence/absence of vanishing point in the previous frame is used to compute the vanishing point in the current frame. The current frame vanishing point is likely to be close to the previous frame. Using this information, a linear motion model for Kalman Filter is constructed to suppress the noise in the vanishing point estimation. The deviation  $(\Delta x, \Delta y)$  in vanishing point from the center of the image corresponds to the change heading angle  $\gamma$ .

$$\gamma = \tan^{-1} \left( \Delta x * \frac{\text{field of view of camera}}{\text{width of image}} \right) \quad (5)$$

The variation in the heading angle ( $\gamma$ ) thus computed is fed to the trajectory planning and control module to recalculate the intermediate way-point, described in next section.

### C. Trajectory Planning and Control

Autonomous flight of the quadcopter following sequence of way-points require fast, real-time trajectory planning

TABLE I: State Estimation Error

Trajectory	x (cm)	y(cm)	z(cm)
Short <30m	92	102	12
Long >30m	96	105	12

and control command generation. The GPS way-points provided by a farmer are converted into local reference frame and provided as input to trajectory planning and control module. Moreover, quadcopter used in the experiment is described by six degrees of freedom of the rigid body as  $Q = [x, y, z, \theta, \phi, \psi]^T$ , where the tuple  $(x, y, z)$  represents the position of the center of mass of quadcopter and roll-pitch-yaw  $(\theta, \phi, \psi)$ , set of Euler angles which represents the orientation in the same reference frame. The proposed trajectory planning and control module generates minimum time trajectory from the current way-point  $(x_w, y_w, z_w)$  to the next way-point  $(x_{w+1}, y_{w+1}, z_{w+1})$  using convex optimization [17]. The trajectory is recalculated from the current position in every update cycle of the navigation framework. This closed loop control allows the framework to handle unplanned drift from planned trajectory during the motion of the quadcopter. Optimum trajectory generation is presented as an optimization problem:

$$\begin{aligned} \text{Minimize} \quad & \Omega = |V_{max} - v_{tk}|^2 \\ \text{Subject to:} \quad & v_{tk} \leq V_{max}, \quad a_{tk} \leq A_{max} \text{ and} \\ & j_{tk} \leq J_{max}, \quad \forall k = 0, \dots, n \end{aligned}$$

where  $v_{tk}$ ,  $a_{tk}$ ,  $j_{tk}$  are instantaneous velocity, acceleration and jerk of quadcopter at time  $tk$  and  $V_{max}$ ,  $A_{max}$ ,  $J_{max}$  are the maximum velocity, acceleration and jerk constraints, details in [17]. The trajectory generation and planning is carried out using generic parameters-acceleration ( $\ddot{x}$ ,  $\ddot{y}$  and  $\ddot{z}$ ) whereas the quadcopter control parameters are  $(\theta, \phi, \psi)$ . Therefore a numerical average of the  $(\ddot{x}$ ,  $\ddot{y}$  and  $\ddot{z}$ ) over a typically small interval (determined experimentally) is computed in each update cycle and transformed into control commands:

$$\phi = \frac{\arctan(\ddot{x} \cos \psi + \ddot{y} \sin \psi)}{\ddot{z} + g} < 90^\circ \quad (6)$$

$$\theta = \frac{\arctan(\ddot{x} \sin \psi - \ddot{y} \cos \psi)}{\ddot{z} + g} < 90^\circ \quad (7)$$

Where  $\theta$  and  $\phi$  are motion commands along forward and lateral direction along with  $\ddot{z}$ , which is another control command for height. The proposed navigation framework works with an assumption of fixed yaw ( $\psi = 0$ ). The inter-row path may slightly curved as discussed in the previous section, hence to avoid drone crash into plantation rows, the quadcopter is required to maintain its path along the middle of the inter-row path while moving towards  $(x_{w+1}, y_{w+1}, z_{w+1})$ . The heading angle  $\gamma$  generated during vanishing point detection in each control update cycle provides direction of middle of the inter-row path. This is utilized for re-calculating  $(x_{w+1}, y_{w+1}, z_{w+1})$ , which is used as a new destination of quadcopter to ensure  $(\theta, \phi, \ddot{z})$  generated in next update, positions the quadcopter towards the middle of the path.

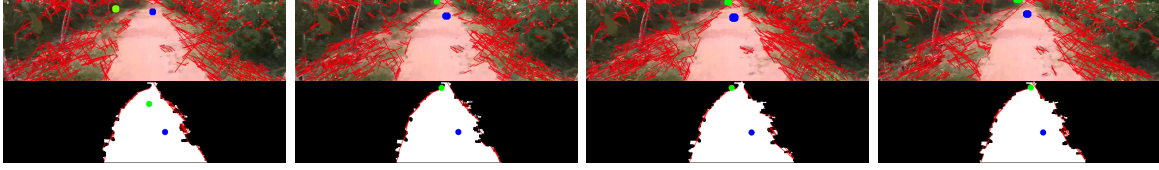


Fig. 8: Vanishing point computation with and without GMM based path segmentation. (Above) - Vanishing point (VP) computed directly on original image with (Red lines) depicting computed candidate lines. (Below) - Vanishing point computed on path segmented image. (Green circle) VP of current frame and (Blue circle) depicting VP after Kalman Filter.

TABLE II: Quantative Results of Autonomous Navigation

Navigation Results	Row1	Row2	Row3	Row4	Row5	Row6	Row7	Total
Success	3	3	4	2	3	3	4	22
Failure	1	1	1	2	0	0	1	6

$$D = \sqrt{(x_{w+1} - x_w)^2 + (y_{w+1} - y_w)^2} \quad (8)$$

$$\begin{bmatrix} x_{w+1} \\ y_{w+1} \\ z_{w+1} \end{bmatrix} = \begin{bmatrix} \sin \gamma & 0 & 0 \\ 0 & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} D \\ D \\ z_w \end{bmatrix} \quad (9)$$

#### IV. EXPERIMENTS AND RESULTS

We have evaluated our proposed framework on Bebop quadcopter [8] with 180° field of view ‘fish eye’ (wide angle) camera. The captured image stream by ‘fish eye’ camera has barrel distortion which is auto-rectified. The rectified image are digitally oriented using ‘tilt-pan’ feature of the camera to get a (wide-angle) view. Frames at 640 × 468 resolution along with orientation information are transmitted to the host device in real time, while frames around (0°) orientation are used for navigation. Video at 1920 × 1080 is stored in the quadcopter and later applied for yield estimation.

Experiments were conducted in an organic pomegranate plantation with over 400 plants and around 2000 ripened fruits at the time of experiments. For all the experiments state estimation, trajectory generation and other purposed framework computations are carried out on a conventional desktop computer running ROS (Robot Operating System) as middle-ware with Ubuntu 14.04 LTS with an Intel Core i5 processor @3.2 GHz and 8GB of RAM. Antecedent to autonomous navigation of quadcopter, accuracy and robustness of the state estimation was established. The quadcopter was flown in the close path of approx 300 meters and the state estimation results were compared with ground truth. Fig. 7 depicts the ground truth, IMU and ORB SLAM fused state estimations along the path. The state estimation errors shown in Table I. The quadcopter was flown in 7

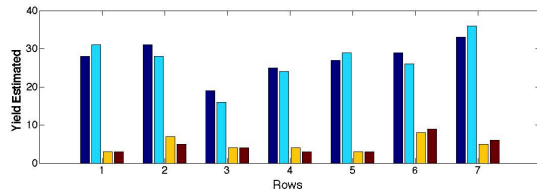


Fig. 9: Pomegranate yield Estimation: Experiments conducted in 7 rows of plantation with manually counted (dark blue) pomegranate fruits yield and (light Blue) determined by our framework. (light brown) manually counted flower buds in plantation and (dark brown) counted by our algorithm.

TABLE III: Yield Detection Baseline

Yield Estimation	Max (Manually)	Min (Manually)	Average (Manually)	Our Process
Pomegranate	72	47	64	69
Flower Buds	37	19	29	25

rows of pomegranate plantation multiple times to establish the navigation framework robustness. As shown in Table II the quadcopter was able to reach its destination without crashing with a success rate of 78.2%. The overall system response time of Bebop quadcopter is about 400 milliseconds with the execution time for navigation framework of 300 milliseconds, which makes it suitable for our application.

Pomegranate and flower yield estimation is baselined manually before comparison with our proposed work. This was done by selecting small set of images at random from dataset and provided to a group of 25 people unrelated to our work for detecting fruits and flower buds manually. On similar set of images our framework was executed and resulted in similar yield estimates, as shown in Table III, where first three columns depict (maximum, minimum and average of all manual detection by 25 people). The quadcopter autonomously navigated through multiple pomegranate plantation rows with trajectory for two rows shown in Fig. 1 and results of yield estimation are shown in Fig. 9. The overall accuracy achieved for pomegranate fruit yield is 89.9% and 88.4% for flower buds.

#### V. DISCUSSIONS

Our path segmentation scheme uses GMMs to model the intensity distribution. The strength of this scheme lies in fast and accurate computation leading to significant better path segmentation. Fig. 10 demonstrates the variations in outputs with changes in the number of Gaussians. Experimentally, we find that bi-model Gaussian is sufficient to model the intensity distribution for path segmentation. The module runs at 5 fps when implemented serially on 3.2 GHz CPU. Fig. 8 illustrates the performance improvement in finding ‘Vanishing point’ (used for navigation) with GMM based path segmentation. We gain considerable performance improvements with ~ 15 times decrease in computation time for computing vanishing points because of less number of computed candidate lines in path segmented image. In comparison to direct VP computation on the original image, we obtain less false variations in the vanishing point position using our path segmentation approach.

The challenges in detection of pomegranates and flowers include occlusions by other objects and limited field of view





Fig. 10: GMM based path segmentation: Qualitative results of path segmentation using 2,3 and 5 gaussians (L to R) respectively. Both cases of undersegmentation and oversegmentation could result in noisy path segmentation. A justified selection of number of gaussians is desired to guarantee correct VP computation.



Fig. 11: Some success and failure cases of pomegranate and flower detection. First two images represent success cases whereas others two images are the failure cases.

of the camera. Shapes which are occluded by the leaves, twigs and other foreign objects are extremely challenging to detect. Even in case of humans counting the desirable objects (as depicted in Table III) we see significant differences in the reported counts. We thus require special care in building a robust solution. Fig. 11 depicts some the challenging cases. The output (count of pomegranate and flowers) of the present frame depends on the previous frames hence we have modeled our system as an order- $n$  Markov. We display the count of ‘objects’ after every  $n$  frames. With an increase in order ‘ $n$ ’ of the system, the complexity grows making our system slow without any significant improvements in the results. Our system performs best in the case of  $n = 3$ . Since, we work with the video as an input to our system, a missed or an incorrect detection at frame level is compensated by taking average over ‘ $n$ ’ previous frames.

The system response time of the quadcopter had to be determined experimentally since internal software architecture is undocumented and wireless communication channel has unpredictable lag. The quadcopter is unstable in high wind conditions which is handled (to certain extent) by trajectory planning and control module due to its closed loop control. Our proposed navigation framework has been designed with fixed yaw ( $\psi = 0$ ) due to unavailability of information and software interface related to dynamic and kinematic constraints of the quadcopter from the manufacturer.

## VI. CONCLUSION AND FUTURE WORK

A novel approach which performs monitoring and yield estimation for pomegranate plantation is presented. An efficient and robust autonomous navigation framework generating minimum time trajectory between sequence of GPS waypoints using convex optimization was developed over ROS. The paper demonstrates the performance of the proposed approach on the pomegranate plantation, but it has promising capabilities applicable to varied plantations. Pomegranate dataset along with navigation framework which would facilitate further research are being delivered to Open Source. Further extensions include integration of Near Infrared Sensor (NIR) over quadcopter to perform more diverse and accurate predictions along with livestock monitoring in plantation.

## ACKNOWLEDGEMENT

This work was supported by grants received from DeitY, National Program on Perception Engineering, Phase-II.

## REFERENCES

- [1] Roberto Benedetti and Paolo Rossini. On the use of ndvi profiles as a tool for agricultural statistics: the case study of wheat yield estimate and forecast in emilia romagna. *Remote Sensing of Environment*, 1993.
- [2] Chia-Che Hung, John Nieto, Zeike Taylor, James Underwood, and Salah Sukkarieh. Orchard fruit segmentation using multi-spectral feature learning. In *(IROS)*. IEEE, 2013.
- [3] John F Federici, Robert L Wample, David Rodriguez, and Suman Mukherjee. Application of terahertz gouy phase shift from curved surfaces for estimation of crop yield. *Applied Optics*, 2009.
- [4] Stephen R Dunn, Gregory M. Yield prediction from digital image analysis: A technique with potential for vineyard assessments prior to harvest. *Australian Journal of Grape and Wine Research*, 2004.
- [5] Cihan Akin, Murvet Kirci, Ece Olcay Gunes, and Yuksel Cakir. Detection of the pomegranate fruits on tree using image processing. In *Agro-Geoinformatics (Agro-Geoinformatics)*. IEEE, 2012.
- [6] Savvas A Chatzichristofis and Yiannis S Boutalis. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Computer vision systems*. Springer, 2008.
- [7] Philipp Krähenbühl and Vladlen Koltun. Geodesic object proposals. In *Computer Vision—ECCV 2014*. Springer, 2014.
- [8] Parrot. Bebop drone. <http://www.parrot.com/products/bebop-drone/>.
- [9] Qi Wang, Stephen Nuske, Marcel Bergerman, and Sanjiv Singh. Automated crop yield estimation for apple orchards. In *Experimental Robotics*. Springer, 2013.
- [10] Debadeepta Dey, Lily Mummert, and Rahul Sukthankar. Classification of plant structures from uncalibrated image sequences. In *Applications of Computer Vision (WACV)*. IEEE, 2012.
- [11] Alison B Payne, Kerry B Walsh, PP Subedi, and Dennis Jarvis. Estimation of mango crop yield using image analysis—segmentation method. *Computers and electronics in agriculture*, 2013.
- [12] AD Aggelopoulou, Dionysis Bochtis, S Fountas, Kishore Chandra Swain, TA Gemtos, and GD Nanos. Yield prediction in apple orchards based on image processing. *Precision Agriculture*, 2011.
- [13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*. IEEE, 2005.
- [14] Raul Mur-Artal, JMM Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *arXiv preprint*, 2015.
- [15] Lucas Vago Santana, Alexandre Santos Brandao, and Mario Sarcinelli-Filho. Outdoor waypoint navigation with the ar. drone quadrotor. In *Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015.
- [16] Georg Klein and David Murray. Parallel tracking and mapping for small ar workspaces. In *ISMAR*. IEEE, 2007.
- [17] Kumar Bipin, Vishakh Duggal, and K Madhava Krishna. Autonomous navigation of generic monocular quadcopter in natural environment. In *Robotics and Automation (ICRA)*. IEEE, 2015.
- [18] Rafael Grompone von Gioi, Jeremie Jakubowicz, and Morel. Lsd: A fast line segment detector with a false detection control. *PAMI*, 2008.