Moving Obstacle Avoidance for an Unmanned Aerial Vehicle

[1]Mrs.Vijayalakshmi B, [2]Divyavizhi A, [3]Gomathi M,[4] Indra S, [5]Kalachithra B
[1]AP ,Department of Computer Science and Engineering ,Ramco Institute of Technology
Rajapalayam, India
[2345]Final Year CSE ,Department of Computer Science and Engineering ,Ramco Institute of Technology
Rajapalayam, India

Abstract *— Tremendous improvement in applications of Unmanned Aerial Vehicle creates popularity in surveillance. In crowded airspace, there is a chance of collisions for UAV. To forge UAV to sense both static and dynamic obstacle, there is a need to emerge a collision avoidance system for to flight safety to reach the target. The crucial thing is to examine both obstacles and UAV movement in airspace using moving camera. This paper presents about the design and implementation of multiple moving obstacles recognition and avoidance by an UAV. Initially, detection approach is done by finding the strong corner points in every frame. Extraction of actual obstacle from the video dataset is obtained by combining the background subtraction and optical flow method. Here, tracking of moving object is done by Kalman filtering technique which deduces noise. Then, predict the obstacles trajectory of UAV and generate a new feasible trajectory for UAV using closed loop RRT technique to avoid collision from other flying vehicles. The implementation is validated using video datasets which is taken from moving camera which is mounted on UAV. The results show that the collision free trajectory are computed fast and satisfies dynamic feasibility.*

Keywords- Unmanned Aeria Vehicle, Object detection

I. Introduction

An unmanned aerial vehicle (UAV) extends their applications from aerial targets and reconnaissance for military purpose to many different civilian areas such as aerial photography, rescue, agriculture, cargo transport. The main challenge for an UAV to fly on its trajectory without any interruption. Heavy lidars are able to detect the distance to the obstacle, but to carry them in an UAV will significantly increase the payload. A monocular camera has been used to detect and tracking moving obstacles, but it relies heavily on the image processing and computer vision algorithms in detection and tracking. Different techniques are emerged for collision avoidance for an UAV. But to develop a reliable algorithm for this purpose is still a research problem till today.

Multiple UAVs can perform various missions. To make UAV run automatically and make decision when the connection between UAV and control system is lost, reliable methods are needed to control the vehicle. Mostly dynamic obstacle avoidance methods include Velocity Obstacle based methods [1]–[3] and Optimal Control based methods [4], [5].

Some challenges associated with dynamic object avoidance algorithms involve the following: (1) Moving obstacle need to be detected even in the far distance to enable timely warning before collision. (2) Motion prediction of moving obstacle need to be accurate. (3) Generation of real time path planner for UAV. (4) Design of execution loop for UAV requites the faulty prediction of obstacles movement.

To solve this problem, methods discussed in this paper can able to predict moving obstacle motion and avoid it under the fast moving camera which is mounted on UAV. First, sequences of frames are parsed from video. Moving object is recognized by differentiating the frames. Because of moving camera, background image is not constant. So, background motion image is estimate by perspective transformation model and then it is fit into global transformation model. Then find the corner points and optical flow in the background subtracted image by Shi-Tomasi [6] and Lucas-Kanade optical flow [7] algorithm. Multiple objects are obtained and it is filtered using Extended Kalman filter algorithm [8] which avoids miss–detections. By knowing the state of a moving object, collision is estimated. Then Closed-loop rapidly exploring random tree [9] provides a path planner which generates a new trajectory when UAV and estimated obstacle path interferes. It is done by generating collision free paths. Each path is checked against constraints, if it satisfies then it added as candidate path. Intermediate way points also generated and added in to path only if satisfies constraints. The above process is repeated whenever collision occurs. The remaining paper discussed briefly about the methods to avoid collision.

II. RELATED WORK

Many approaches are evolved for collision avoidance of UAV. Some detection technique like frame differencing detect object by differencing the pixels between frames. But it is work well in static background. Also, few techniques need some specialized hardware to implement and it increases the payload of UAV. Moving object can be identified quickly using extraction of corner points in each frame. Susan corner detection rely on brightness of pixel, fixed global threshold does not suit for general situation. Anti-noise and robustness of this algorithm is also weak. Markov Decision process based methods sense unreliability, but it cannot able to tackle when number of obstacles is large. Different methods developed to avoid obstacles like potential filed, cross entropy, elliptical search, but it works only in static. In challenging environment, obstacles are large and far and also covered with noise. So, there is a need to develop reliable techniques to alert the system before occurrence of collision.

Mostly, many techniques rely on moving obstacle motion for avoidance. Estimation of small moving object is identified using frames variation and noise elimination. In this paper, background subtraction method and optical flow are combined to detect object under moving camera with performance improvement. In Background subtraction method, background motion estimation is needed for fast moving camera which will helps for foreground detection. Optical flow matching of background subtracted image can extract spatio-temporal features of moving object. This helps to track moving objects. To track objects kalman filter is used, but it is used for linear systems. For non-linear systems or virtual reality applications, extended kalman filter (EKF) is best than previous. EKF paves the way to estimate state of object. Closed-loop Rapidly exploring random tree create smooth trajectories than RRT and also deduce the prediction error.

III. DESCRIPTION OF METHODS

As illustrated in Fig.1, we propose an efficient moving obstacle avoidance algorithm for UAV's.

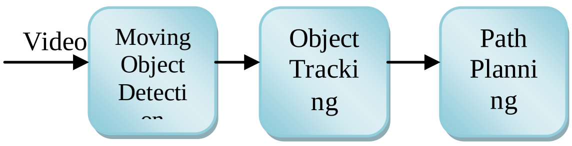

Fig.1. An overview of our proposed method

We first detect the moving obstacle from video. By using the result obtained from detection track the objects. Additional moving object detection operations are performed to differentiate target objects from spurious noise. Finally if there is any obstacle in path, finds new path by path planning.

We use spatiotemporal characteristics of each detected object to identify actual UAV and incorporate the temporal consistency of detected objects through tracking. In the following section, we describe each component of our algorithm in detail.

1. *Moving Object Detection*

The different approaches are suitable for characterizing small moving objects since their motions can be estimated in local regions between frames. Motion-based approaches are divided into two main categories:

Fig.2. Extraction of frames from video data set

a) Background Motion Estimation:

For background motion estimation, we are going to assume that the background moves smoothly. From a sequence of video frames (Fig. 2), we extract a set of corner points (Fig. 3(a)) and estimate local motion fields (Fig. 3(b)) on those selected corner points. The computed local motion fields are then transformed into a global transformation which represents the background motion.

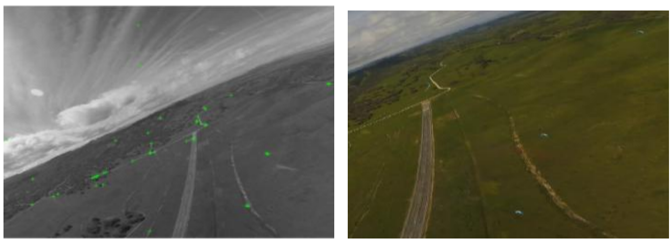

Fig.3. Snapshot of corner points in the frame (Figure 3(a)). Illustration of optical flow among moving objects in a frame Figure 3(b).

i) Identify Corner Points:

First we have to find corner points from a video frame. We are using shi Tomasi corner detector also called as minimum eigen value method, due to effiency,speed and finding of strongest corner points. Shi Tomasi corner detector is based on assuming the corners are associated with the local autocorrelation function.

Given a frame A, we define the local auto-correlation function C at pixel P as following.

$$C(P)= \sum_{w}[\nabla A(s)\cdot \delta s]^2$$
$$= \delta s^T \sum[\nabla A (s)^T \nabla A (s)]\delta s$$
$$= \delta s^T \Lambda \delta s$$

where ∇A is the first order derivative of the image and Λ is the precision matrix.

In Shi-Tomasi corner detection, a corner Q is calculated to eigenvalues of Λ

$$Q(s) = \min\{\lambda_1, \lambda_2\}$$

where λ_1 and λ_2 are eigenvalues of Λ.

After thresholding on Q, find a set of salient points. Discard points for which there is a stronger corner points at a certain distance.

ii) Using Corner Points, Finding Local Motion Field:

First step is finding the local motion field from the previous frame F_{t-1} to the current frame F_t on the identified corner points (Fig. 2). Then we take c_{t-1} as one of the Shi-Tomasi corner points in F_{t-1}.After that the motion vector is computed as M_t from the point c_{t-1} using Lucas-Kanade method [] by assuming that our local motion is optical flow.

In Lucas-Kanade method, all the points which are placed around the given pixel must have the same motion. So the local motion can be predicted by solving the least square problem.

$$M_t = \arg\min_{M} \sum_{s\in N(c_{t-1})} | X_t(s+M) – X_{t-1}(s)|^2$$

Where $N(c_{t-1})$ is the neighbourhood around the c_{t-1}. The above equation is easy to solve with the closed form solution. Further we use the bi-directional verification to obtain the accurate motion vector field such that $\|M_t + (M_t)^{-1}\|^2$ has small value.

iii) Local Motion Field Fitting into Global Transformation:

After local motion fields M_t is founded then fit them into global transformation. By using the optical-flow matching the corresponding point c_t in the current frame F_t is represented as $c_t = c_{t-1}+M_t$.

After finding the global transformation G_t, this G_t regularizes the local motion field to be smooth in the entire image.

$$G_t = \arg\min_{G} \sum_{c_t \in C_t,\, c_{t-1} \in Ct_{-1}} \| c_t – G \circ c_{t-1}\|^2$$

Where C_t and C_{t-1} represents a set of corresponding points in current frame F_t and F_{t-1} and o is the warping operation.

After, we have lots of global transformation model such as rigid or affine transformation model. Then we use the perspective transformation model which reflects the UAV occupy a small portion of the field of view. For computation we use a efficient model as perspective transformation which requires only 9 parameters to describe and it takes a projection which is based on the distance from the camera. At last the result of the perspective transformation in the above equation is assigned to the background motion between two consecutive frames.

b) Computation Of Subtracted Image:

By using the estimated background motion, background subracted image will be computed which highlight the moving objects that have more complex motion. After that we identify the corner points in the background subtracted image and local motion vector on those points to be found using the appearance information. By assuming that the motion of the target object is greatly different from the background motion we perform the additional test to prune the spurious noise.

i) Background Subtracted Image Computation:

We can subtract the background by using the original image and the background motion compensated image. If the background motion estimation may not be accurate as the assumption of the single plane on the perspective model can be violated in the video. For obtaining a more accurate background subtracted image, we use the estimation of background motions from the multiple previous frames.

Then the perspective transform between the two previous frames from F_{t-2} to F_{t-1} can be denoted as G_{t-1}. After that by taking the average of forward and backward tracing, the background subtracted image B_{t-1} for X_{t-1} is computed.

$$B_{t-1} = 1/2\, |F_{t-1} – G_{t-1} \circ F_{t-2}| + 1/2\, |F_{t-1} – (G_t)^{-1} \circ F_t|$$

Where the inverse transform of G_t is $(G_t)^{-1}$. Here the background subtracted image is computed for the previous frame F_{t-1} for symmetry.

ii) Find the Corner Points on Moving Objects:

In the background subtracted image B_{t-1}, the moving objects are highlighted. For detecting the moving object in the current frame we have to find the corresponding regions in F_t. For this we have to extract Shi-Tomasi corner points in B_{t-1} and propagate them to appearance image F_{t-1}. By using the Lucas-Kanade method we need to find the corresponding point in F_t for each propagated corner point from F_{t-1}.

By denoting the corner point as Q_{t-1} in F_{t-1} propagated from B_{t-1}. Then the local motion field L_t is computed as,

$$L_t = \arg\min_{L} \sum_{s \in N(P_{t-1})} |F_t(s + L) - F_{t-1}(s)|^2$$

Where the neighbourhood around P_{t-1} is $N(P_{t-1})$. It's worth noting that we use the appearance image to estimate the local motion not using the background subtracted image.

iii) Remove Salient Points based on Motion Difference:

For detecting the moving objects we have the corresponding points in F_t from B_t. Due to the incorrect background motion estimation there is a points on spurious noise (i.e. edge of background).

For that we need to remove the points based on the difference between the estimated background and local motion. This happens based on the assumption of target object has different motion compared with background.

The motion difference D_t between the background and moving object as follows:

$$D_t = g_t - L_t$$

Where interpolated motion vector is g_t from the perspective transformation G_t at the point P_{t-1}. Then based on the magnitude of motion difference we find the pruned point R_t.

$$R_t = P_{t-1} + L_t \quad \text{if } \|D_t\|^2 > T$$

Where empirical threshold for pruning is T .Then the

Pruned points are cluster according to spatial proximity. By generate the bounding box for each cluster of points which represents our detection for a single moving object.

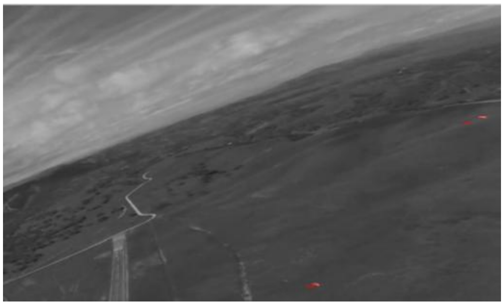

fig.4. Illustrates about the detection of moving object which is indicates by red bounding box.

2. Estimation Of Moving Obstacle State:

To estimate the collision between UAV and obstacle, we have to find the position and velocity of the moving obstacle. For this estimation, extended kalman filtering technique is used. This filtering technique is required to neglect the miss-detections.

This filter analyse the previous state P_{t-1} and estimate the current state P_t by adding the current value C_t with current state as,

$$P_t = TP_{t-1} + S_t$$

$$C_t = MP_t + E_t$$

Where, T – transition control matrix,

S_t - controls transition error

M – measurement matrix

E_t - measurement error

Then, the current output P_t is estimated using kalman gain K_G.

$$P_t = TP_{t-1} + K_G(C_t - MP_t)$$

$$K_G = V_S M^T (MR_sM^T + V_E)$$

Where, V_S & V_E are covariance of S_t & E_t.

Using state P_t, we can obtain the position and velocity of the object by assuming constant velocity model on transition control and measurement matrix. In L previous frames, particular detected object (Fig. 4) is identify using optical flow matching and check for consistent. By using kalman output, correct the miss-detection in current frame.

3. Path Planning:

To avoid collision of UAV, we can use Motion planning in dynamic environments. For autonomous driving, the Closed-Loop RRT is developed.

To avoid dynamic obstacles we use a chance constraint RRT for planning a path. So a sampling based path planning method is proposed for UAV dynamic obstacle collision avoidance because of the following advantages: 1) Able to find a feasible motion plan in a short span of time. 2) Guarantees probabilistic completeness. 3) Applicable to different dynamic models. 4) Enables trajectory-wise checking.

Our algorithm is shown in the following steps: 1) Sample a point from 3D UAV work space. 2) Use the UAV's closed loop system to simulate a trajectory of the UAV and a linear motion model to simulate trajectories of dynamic obstacles and identify potential collisions. 3) Store collision free trajectories as candidate paths. After repeating this cycle for fixed amount of time, the shortest trajectory is chosen from the list of collision free candidates. This process is repeated at the next time step if collisions are identified with the previous generated collision free trajectory.

Constraints:

1) Dynamic obstacle should not enter into UAV's safety zone to ensure collision free.

2) Height limits and a geo-fence, must also be enforced.

a) The Closed-Loop RRT Based Method:

There are two main types of sampling-based methods: graph based and tree based planners. Graph based method uses random sampling in the state space to build a roadmap of the free state space. It can also be used to create paths from the roadmap. Tree based methods has a vertex selection method, which determines expansion among vertices in the graph. After that, a local planning method constructs an edge from the selected vertex, thereby extending the tree. Closed-Loop Rapidly Exploring Random Tree (CL-RRT) is chosen from tree-based planners.

Algorithm 1- TreeExpand()

1: for Δt do
2: Generate a sample z_s.
3: Sort the nodes in the tree by heuristics in ascending order.
4: **for** each node q in the tree, in the sorted order **do**
5: Simulate a trajectory $\chi(t)$, t € [t1, t2] from q to z_s using the closed-loop system and check the trajectory against constraints in the above mentioned.
6: **if** Satisfy the constraints **then**
7: Add the end of X(t) to the tree with q as their parent. Break.
8: **end if**
9: **end for**
10: **for** each newly added node q_n **do**
11: Calculate the cost of q_n by adding length of the trajectory from its parent and to the cost of its parent.
12: Simulate a trajectory from q_n to the goal using the closed-loop system and check it against constraints which is mentioned above.
13: **if** Satisfy the constraints **then**
14: Mark q_n as goal reachable.
15: Calculate cost-to-go of q_n as the length of the trajectory to the goal.
16: **end if**
17: **end for**
18: **end for**

First, the nodes in the tree are sorted in an ascending order by using heuristics (line 3). Heuristics is the approximate cost from each node to the sample node. The node q with the smallest heuristics is chosen and a trajectory from that q to the sample node z_s is simulated using the closed-loop system. Then the trajectory is checked against the constraint which is stated above. If the constraints are satisfied, the node q is added to the tree (line 7). For this newly added node q_n, its cost is calculated by adding the trajectory length from its parent q to q_n and the cost of q. Then an attempt is made to reach from q_n to the goal: a trajectory from q_n to the goal is simulated using the closed-loop system and checked against constraints (line 12). If it satisfies the constraints, q_n will be marked as reachable to the goal and its cost-to-go is the length of the simulated trajectory. To generate a path, the goal- reachable node $q*$ with smallest sum of cost and cost-to-go is selected. The node sequence connecting the start to the goal via $q*$ is the generated path.

b) Execution Loop:

In each loop, the algorithm predicts a collision between the UAV and the obstacle aircraft by simulating the UAV's trajectory up the time horizon Δt_m. Δt_m is determined by the time needed to make a turning movement for the UAV to avoid collision. If a collision is predicted, the path planners will help in generating an avoidance path. Otherwise a reactive avoidance action such as climbing up will be taken. Once a path is generated it' not immediately sent to UAV. Rather, a new trajectory is simulated using the closed-loop system along the waypoints representing the path. This waypoint is checked against constraints with updated states of the UAV and moving obstacles. If the constraints are satisfied, the waypoints will be sent. Otherwise, the planner needs to create a new path for collision avoidance. The loop is repeated until the UAV reaches its goal.

Algorithm 2 - ExecutionLoop()

1: **repeat**
2: Predict the collision up to a time horizon Δt_m
3: **if** A collision is predicted **then**
4: **if** Collision location is close to the start location **then**
5: take reactive avoidance action.
6: **else**
7: TreeExpand()
8: **if** No path is feasible **then**
9: Go to line 7
10: **end if**
11: Update the UAV and the obstacles' states and simulate a new UAV trajectory starting from the updated position to the goal and check against the constraints.
12: **if** Constraints violated **then**
13: Go to line 7
14: **else**
15: Send the path to UAV.
16: **end if**
17: **end if**
18: **end if**
19: until Finish the waypoint sequence

IV. CONCLUSIONS

In this paper, we proposed a method for Unmanned Aerial Vehicle to avoid moving obstacles. Our method first finds Shi- Tomasi corner points and Lucas-Kanade optical flow among frames. Then estimate the Background Compensated image by fitting the local motion vectors into global transformation. Next, use the background subtraction method to detect moving objects. Then filter and track the objects using kalman-filter technique to avoid spurious noise. Closed loop RRT based path planner is used to develop a reliable avoidance path. In this planner the generated way points and trajectories are able to avoid collisions. Using reachable set, prediction mismatch is avoided. This algorithm can be used to avoid large number of obstacles with high accuracy.

V. REFERENCES

[1] Siyao, F., L. Han, Yu Tian, and G. Yang, *"Path planning for unmanned aerial vehicle based on genetic algorithm"*, 2012 11th IEEE International Conference on Cognitive Informatics, 22-24 Aug. 2012.

[2] Yoshiaki Kuwata, Justin Teoy, Sertac Karamanz, Gaston Fiorex,Emilio Frazzoli, and Jonathan P. Howk,*"Motion Planning in Complex Environments using Closed-loop Prediction",*

[3] T. McMahon, S. Thomas, and N. M. Amato, *"Sampling based motion planning with reachable volumes"***,** Citeseer, Tech.Rep,2013.

[4] Yucong Lin and Srikanth Saripalli, *"Collision Avoidance for UAVs Using Reachable Sets"*, 2015 International Conference on Unmanned Aircraft Systems, June 9-12, 2015.

[5] Yucong Lin and Srikanth Saripalli, *"Path Planning Using 3D Dubins Curve for Unmanned Aerial Vehicles"*, 2014 International Conference on Unmanned Aircraft Systems, May 27-30, 2014.

[6] Qiang Li, Ranyang Li, Kaifan Ji, Wei Dai, *"Kalman Filter and Its Application"*, 2015 8th International Conference on Intelligent Networks and Intelligent Systems.

[7] WANG Yang, CHEN Wenjie**,** *"Path Planning and Obstacle Avoidance of Unmanned Aerial Vehicle Based on Improved Genetic Algorithms"*, Proceedings of the 33rd Chinese Control Conference July 28-30, 2014.

[8] D. Bareiss and J. van den Berg, *"Reciprocal collision avoidance for robots with linear dynamics using LQR-obstacles"*, in *Proc. IEEE Int.Conf. Robot. Autom*, May 2013, pp. 3847–3853.

[9] P. Fiorini and Z. Shiller, *"Motion planning in dynamic environments using velocity obstacles"*, *Int. J. Robot. Res.*, vol. 17, no. 7,pp. 760–772, 1998.

[10] J. van den Berg, D. Wilkie, S. J. Guy, M. Niethammer, and D. Manocha, *"LQG-obstacles: Feedback control with collision avoidance for mobile robots with motion and sensing uncertainty"*, in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2012, pp. 346–353.

[11] D. H. Shim and S. Sastry, *"An evasive maneuvering algorithm for UAVs in see-and-avoid situations"*, in *Proc. Amer. Control Conf. (ACC)*, Jul. 2007, pp. 3886–3891.

[12] Yao-hong, Q., P. Quan, and Y. Jian-guo, *"Flight path planning of UAV based on heuristically search and genetic algorithms"*, Thirty-First Annual Conference of the IEEE Industrial Electronics Society, 6-10 Nov. 2005.

[13] Siyao, F., L. Han, Yu Tian, and G. Yang, *"Path planning for unmanned aerial vehicle based on genetic algorithm"*, 2012 11th IEEE International Conference on Cognitive Informatics, 22-24 Aug.2012.

[14] S. Karaman, E. Frazzoli.*"Optimal vehicle routing with metric temporal logic specifications", IEEE* Conference on Decision and Control, 2008.

[15] Chris Stauffer and W. Eric L. Grimson, *"Learning patterns of activity using real-time tracking"*, Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 22, no. 8, pp. 747–757, 2000.

[16] Chung-Ching Lin and Marilyn Wolf, *"Detecting moving objects using a camera on a moving platform"*, in Pattern Recognition (ICPR), 2010 20th International Conference on. IEEE, 2010, pp. 460–463.

[17] P. Sand and S. Teller, *"Particle video: Long-range motion estimation using point trajectories",* in Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, 2006, vol. 2, pp. 2195–2202.

[18] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. How, and G. Fiore, *"Real-time motion planning with applications to autonomous urban driving",* Control Systems Technology, IEEE Transactions on, vol. 17, no. 5, pp. 1105–1118, 2009.