

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/301845247>

Autonomous navigation for low-altitude UAVs in urban areas

Article · February 2016

CITATIONS

8

READS

356

5 authors, including:



Thomas Castelli

Université Jean Monnet

2 PUBLICATIONS 12 CITATIONS

[SEE PROFILE](#)



Alain Treméau

Université Jean Monnet

254 PUBLICATIONS 2,230 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Semantic Segmentation using Deep Learning [View project](#)



Colour and Space in Cultural Heritage (COSCH). [EU COST Action TD1201] [View project](#)

Autonomous navigation for low-altitude UAVs in urban areas

Thomas Castelli^{1,2}, Aidean Sharghi³, Don Harper³, Alain Tremeau² and Mubarak Shah³

Abstract—In recent years, consumer Unmanned Aerial Vehicles have become very popular, everyone can buy and fly a drone without previous experience, which raises concern in regards to regulations and public safety. In this paper, we present a novel approach towards enabling safe operation of such vehicles in urban areas. Our method uses geodetically accurate dataset images with Geographical Information System (GIS) data of road networks and buildings provided by Google Maps, to compute a weighted A* shortest path from start to end locations of a mission. Weights represent the potential risk of injuries for individuals in all categories of land-use, i.e. flying over buildings is considered safer than above roads. We enable safe UAV operation in regards to 1- land-use by computing a static global path dependent on environmental structures, and 2- avoiding flying over moving objects such as cars and pedestrians by dynamically optimizing the path locally during the flight. As all input sources are first geo-registered, pixels and GPS coordinates are equivalent, it therefore allows us to generate an automated and user-friendly mission with GPS waypoints readable by consumer drones' autopilots. We simulated 54 missions and show significant improvement in maximizing UAV's standoff distance to moving objects with a quantified safety parameter over 40 times better than the naive straight line navigation.

I. INTRODUCTION

UAVs are becoming increasingly present in our everyday lives, their extensive use recently jumped from military to hobby and professional applications. The consumer market is growing and now it offers a wide range of micro and mini UAVs at affordable costs. But this popularity induces some dangerous behavior, most people do not realize that a simple mistake can cause severe injuries to themselves or others. In the United-States, the FAA has taken measures to inform hobbyists and encourage them to follow a code of conduct to prevent accidents. The only form available is the advisory circular ‘AC 91-57’ from June 9th 1981, it advises pilots to keep their UAVs within their line of sight, below 400 feet above ground level, further than 5 miles from an airport (or warn them), and to avoid flying above people. Even for the vast majority of UAV users that are responsible and careful in their use, there is no automated means to fly safely in regards to the UAV’s environment. Our work aims to provide such functionality to micro and mini UAVs that are operated in urban areas.

In this paper, we propose a novel method for autonomous navigation for low-altitude UAVs in urban areas. For a

given mission our method computes safe waypoints, which dynamically adapt the flight plan to the UAVs surroundings by avoiding objects such as cars and pedestrians. We take advantage of satellite and georegistered data to adapt the UAVs mission layout by computing a weighted shortest path instead of flying in a straight line. Weights in our cost function for computing the flight path are defined using land-use summarized in three classes: most dangerous areas are roads and paths where people are prone to the danger the UAV represents, safest are buildings and water, and the rest is in between (Fig. 1). For increased safety, our method also adapts dynamically to moving objects while in flight by adding new local weight to the global weight map.

In our general scenario, we assume a UAV with video camera flying over a given geographical region, for which geodetically accurate reference image, GIS data of buildings and road networks are available. Captured videos are georegistered with the reference image in order to transfer pixel coordinates to GPS coordinates. Moving objects, e.g. vehicles and pedestrians, are detected and tracked from frame to frame. Given the tracks, GIS and reference image, the optimal UAV path is dynamically computed. For simplicity in this paper, we employ ground truth tracks available from WPAFB and PVLabs datasets providing geo-registered images and ground truth for moving objects. Finally, we simulate a real flight by complying with the ‘AC 91-57’ form and using parameters of compatible hardware.



Fig. 1. Visualization of the weight map overlaid on the corresponding satellite image, for WPAFB dataset. Colors represent costs in the weight map, red, transparent and green respectively represent dangerous, neutral, and safer areas.

¹ Survey Copter / Airbus Defense and Space, Pierrelatte, FRANCE

² Hubert Curien Laboratory, Saint-Etienne, FRANCE
alain.tremeau@univ-st-etienne.fr,
thomas.castelli@univ-st-etienne.fr

³ Center for Research in Computer Vision, UCF, Orlando, USA
aidean.sharghi@knights.ucf.edu,
harper@ucf.edu, shah@crcv.ucf.edu

II. RELATED WORK

Many different topics are studied to enhance the usability and to develop new functions to make drones more capable and autonomous. There are several subfields which are related to this work including video geo-registration, detection and tracking of moving objects in videos, detection of roads, buildings, water bodies from satellite imagery and flight path planning.

The most popular trend in UAV video analysis has been moving object detection and tracking from aerial images, many approaches have been proposed with or without using GIS data and geo-registration steps. Kimura *et al.* [5] use epipolar constraint and flow vector bound to detect moving objects, Teutsch *et al.* [6] employ explicit segmentation of images, Xiao *et al.* [7] restrain the search on the road network, and Lin *et al.* [8] use a motion model in geo-coordinates. Moving object detection and tracking are mainly used to follow targets, for surveillance as Quigley *et al.* [4] and Rafi *et al.* [9] describe with their flight path adaptation solutions, or for consumer applications at very low-altitude as in [18] and [19].

Another area that has been getting a lot of attention is autonomous navigation. Different subproblems have been studied, path planning in dynamic environment [10], [11], GIS-assisted and vision-based localization using either road detection [12], buildings layout [14] or DEM (Digital Elevation Map) [13]. Various methods have been proposed for UAV navigation, using optical flow with [15] or without DEM [16], or using inertial sensors [17].

Obstacle avoidance is also a big concern for automating UAV operation, but research has mostly been focused on ground robots [20], [22], even if there has been adaptations for UAVs as Israelsen *et al.*'s intuitive solution for operators [21].

The approaches for autonomously navigating UAVs have been studied, but previous work focus on target following or keeping the UAV's integrity. However, in this paper we propose an autonomous UAV navigation method in order to increase public safety in regards to drones operation, and also to prevent UAVs finding themselves in difficult situations.

III. OUR METHOD

Our contribution towards safe integration of small UAVs into the airspace has two main steps. The first step, described in section A and B, takes into account the physical surroundings of the UAV by computing, as part of the mission preparation, a global path between the user-given start and end locations. This path is represented as a succession of waypoints, exactly as users are accustomed to in mission planner softwares. Before takeoff the user is able to validate the automated path, and he can modify the waypoints if needed. The second step, described in section C, runs in online fashion during the flight and takes into account the environment of the UAV by dynamically adapting its behavior in regards to moving objects that need to be avoided.

A. Extracting the geo-referenced weight map

A convenient approach to gain awareness of the UAV's surroundings is to use satellite imagery and the meta-data provided by Google Maps, DigitalGlobe, Planet Labs or others. To jointly use geo-registered data and aerial imagery obtained from UAV, there has to be a common representation and space. Public tools providing satellite images are very popular and well integrated in third party UAV software such as Mission Planner. For simplicity and compliance, the solution is then to register video images onto a geo-registered satellite image of the area of interest. UAV and world coordinates systems are related with (1), as described in [2].

$$\vec{X}_{camera} = G_y G_z R_y R_x R_z T \vec{X}_{world} \quad (1)$$

with \vec{X}_{world} the world coordinates system, T is the translation matrix derived from the vehicle's latitude, longitude and altitude, and G_y , G_z , R_y , R_x and R_z are rotation matrices regarding respectively camera elevation angle, camera scan angle, vehicle pitch angle, vehicle roll angle and vehicle heading angle.

We chose to use Google Maps API for it's convenience and for the quality of the data provided¹. This free API allows anyone to request satellite images and roadmaps displaying buildings and roads. These three links, 1, 2 and 3, give example commands to request satellite, road map and building images.

We assume that flying above buildings represents less risk than doing so above other environmental elements such as roads or crowded streets. The resulting weight map for the Wright-Patterson Air Force Base (WPAFB) area, shown in Fig. 1, displays three categories:

- Red is to be avoided, for roads and paths.
- Green is to be preferred, for buildings and water.
- Transparent is in between, for other land-use.

To extract the map, only two GPS locations are needed as input from the operator: top left and bottom right GPS coordinates. A grid of image GPS locations is then computed based on Google Maps' camera parameters and resolution level, in other words the ground sampling distance (GSD), to ensure sufficient overlap between images for stitching. We have defined the GPS grid in a way that successive images have pure translations between them. We thus can stitch them together using straight forward normalized cross correlation, which is a robust and fast method given that we manipulate large images and avoid scale change and rotation. This process allows us to minimize the error while creating the geo-registered map. As a result we obtain 3 images for any given area (Fig. 2). Given the image center GPS location, Lat_1 and Lon_1 , the corresponding GPS location (Lat_2 and Lon_2) of all other pixels location at $(\Delta x, \Delta y)$ from the center, can be determined as follows:

¹The proposed method is not dependent on the source, any satellite image and data provider can be used.

$$Lat_2 = Lat_1 - \sin^{-1}\left(\frac{r \cdot \Delta y}{E_r}\right) \quad (2)$$

$$Lon_2 = Lon_1 + \sin^{-1}\left(\frac{r \cdot \Delta x}{E_r \cdot \cos(Lat_1 \cdot \frac{\pi}{180})}\right) \quad (3)$$

with Lat_1 , Lat_2 , Lon_1 , Lon_2 representing latitudes and longitudes of the start and end points, E_r the mean radius of the earth, r the pixel ratio in meters per pixel depending on the ground altitude, and on the requested image scale and resolution, Δx and Δy are the difference in pixels on the map between the two points.

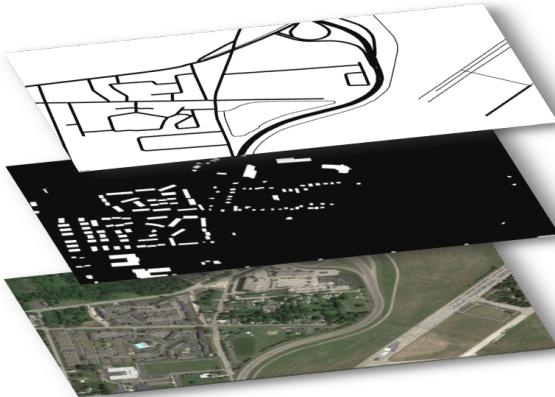


Fig. 2. This figure shows from bottom to top three types of data used in this work: satellite image, buildings and water map, and roads map.

B. Global path planning

The vast majority of UAS (Unmanned Aerial Systems) can be used with a ground control station (GCS), for example APM:Copter (previously known as ArduCopter) has its own mission planner with all the necessary tools. The conventional ways of controlling UAVs are either with a manual radio controller or by using a GCS that defines successive GPS waypoints (specifying the GPS location, altitude, and velocity) to which the UAV will fly autonomously. Despite their efficiency and convenience, there is a crucial flaw with waypoints; they are defined by the user and do not take into account the surroundings of the UAV. This is precisely what we want to tackle with our global path computation. By using the three types of data shown in Fig. 2, we define the optimal path (example in Fig. 3) between two points and thus add a safety parameter to mission planning.

We find the safest route between two GPS coordinates by converting them into image pixels and computing a weighted shortest path algorithm using A* algorithm. The segments' lengths between two adjacent pixels are the Euclidian distance multiplied by the weight defined by the map class. Pixels that are in red have a weight of 100, green is at 5, and the rest is at 20. Those values have been determined empirically. This process ensures that the red areas are avoided but also makes sure the UAV wouldn't take a long



Fig. 3. Different path planning solutions between two GPS coordinates. White dotted path: classic straight line path used in typical systems using waypoints. Blue path: our method, which determines the shortest path by minimizing the cost function such that the resulting path avoids flying over red areas that are dangerous and prefer green areas that are safer.

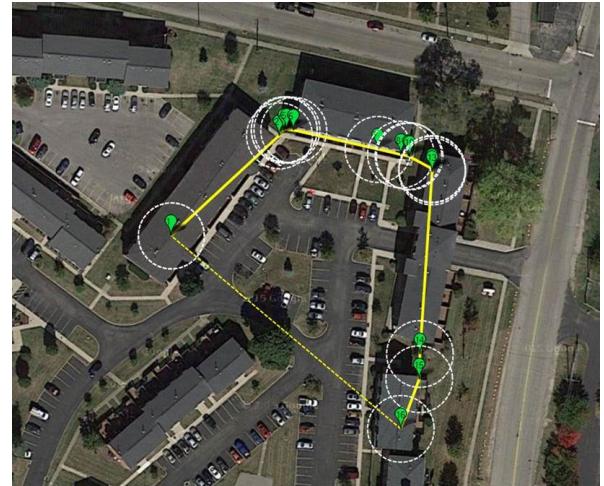


Fig. 4. Path computed by algorithm, shown in yellow, converted to waypoints and visualized in Mission Planner. Green ticks are waypoints locations, white dotted circles are areas where the UAV will consider having reached the waypoint, and yellow dotted line represents the simple straight path.

detour to reach its destination, thus keeping the loss of flight time to a minimum.

As the map is geo-registered, the outputted path can easily be converted into GPS coordinates using (2) and (3), and put in KML or TXT files to be readable by mapping software and ground control stations (Fig. 4).

This global weight map considers the static environment that the UAV will encounter such as roads and buildings. In order to ensure a higher level of safety in all stages of the flight, we also adapt the path locally during the flight in regards to moving objects as explained in the next section.

C. Local path planning

For increased safety, the path needs to be adapted dynamically during the flight to avoid moving objects detected in

the field of view of the UAV's embedded camera. In order to ensure a sufficient distance margin between each object and the UAV, the weight map used for shortest path is modified according to the objects' location, trajectory and velocity. We compute the new weights of the map by applying at chosen locations a multivariate normal probability density function.

The variance Σ_x and Σ_y for each distribution are dependent on the object's characteristics. The x term is proportional to the width of the object in pixels, and y term (4) is proportional to the object's velocity.

$$\Sigma y = V_{Obj} \cdot S \quad (4)$$

where V_{Obj} is the current velocity of the object and S the safety margin to avoid collision in seconds.

The resulting distribution is normalized, rotated to align with the object's trajectory, and centered on the chosen location (5). The weight map is then multiplied with the distribution instead of being swapped in order to keep the global environment based information.

The locations where the distribution is applied to are defined given two criteria. One is whether the object collides with the UAV's path, and the other is how this collision happens. The object and UAV will take respectively t_{Obj} and t_{UAV} seconds to the collision point, if $|t_{Obj} - t_{UAV}| < \Delta$ (Δ is set to 5 s in our experiments), the distribution is applied on the collision point and also on a projected location to avoid re-planning a path that will create a similar situation. The projected location (6, 7) is estimated as follows: the time for the UAV to travel to the current object's location is computed, the projected location is where the object will be at that time given constant velocity and trajectory for the object. For objects that will not collide or that do meet the requirement of $\delta t > \Delta$, the distribution is applied at the next and projected locations.

$$\Omega = \Omega \cdot [R \ T] \circ \Phi \quad (5)$$

where Ω is the weight map, $[R \ T]$ the affine transformation applied to the multivariate normal probability density function Φ to allocate costs to Ω . Φ is centered at the wanted location $L_{x,y}$ and oriented given the rotation matrix R dependent on the object's trajectory and

$$T = \begin{bmatrix} L_x^{Obj} + D \cdot \sin(\alpha) \\ L_y^{Obj} + D \cdot \cos(\alpha) \\ 1 \end{bmatrix} \quad (6)$$

is the translation component of the affine transformation, L_x^{Obj} and L_y^{Obj} are the image coordinates of the object, and $D \cdot \sin(\alpha)$ and $D \cdot \cos(\alpha)$ are respectively the distances in X and Y to the desired location.

$$D = \begin{cases} V_{Obj} \cdot \frac{d_{UAV-Obj}}{V_{UAV}} & \text{for projected location} \\ V_{Obj} \cdot \frac{1}{fps} & \text{for next location} \end{cases} \quad (7)$$

with D the distance used in 6, V_{Obj} and V_{UAV} the current velocities of the object and UAV, $d_{UAV-Obj}$ the distance

between object and UAV, and fps the frame-rate or computation time.

This method will ensure that the resulting path will leave sufficient ground distance between objects and the UAV, and if multiple objects are close together, it will create a barrier and encourage the UAV to find a safer path, thus preventing it to fly above any moving objects (Fig. 5).

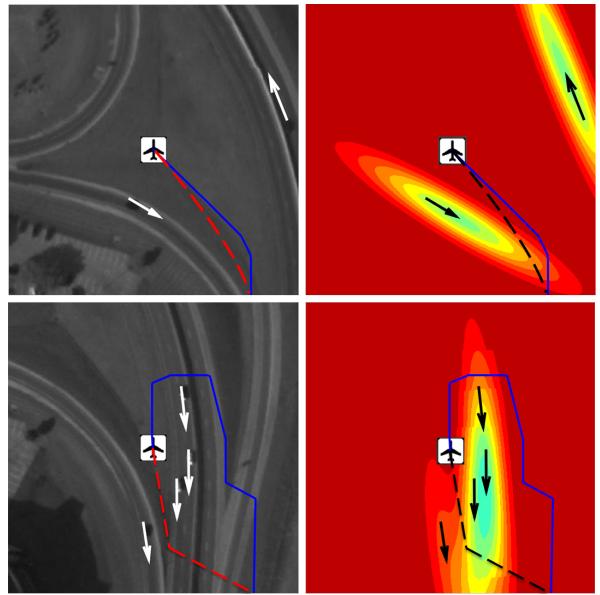


Fig. 5. Left column shows the images and right column shows the corresponding weight maps. Objects trajectories are shown in white (in images) and black (in weight maps). The global paths shown in red and black dashed lines, are adapted with weight adjustments to avoid flying over objects. Please note that, in the bottom row, the path crosses the road perpendicularly on the right part of the images (more visible on the bottom-left image). This is due to the fact that we want to minimize crossing high-cost road pixels.

IV. EXPERIMENTS

A. Methodology

In order to simulate a real world scenario as accurately as possible, our method uses dataset images, and typical UAVs' specifications and camera parameters. We made sure to comply with the latest regulations and advice regarding UAV operation, and used the following flight and hardware parameters:

- Altitude above ground level : 50m.
- Velocity : < 15m/s.
- Camera Horizontal Field Of View (HFOV) : 97.40°².
- Horizontal ground sampling resolution : 8.84cm/pixel.

The principles used to build the simulation scheme are the following:

- UAV videos are registered in the geo-referenced space, we can thus work in pixels coordinates, and convert back to GPS anytime.

²HFOV for a configuration using a PointGrey Blackfly 1,3MP 1/3" camera of 1288x964 resolution and a Kowa LM3PB lens.

- The datasets' ground truth gives the moving objects' location for every frame (motion vectors in Fig. 6).
- The UAV will follow the global path (blue in Fig. 6).
- For every frame the UAV's displacement in the image is dependent on its velocity and direction (8).
- The considered objects are only the ones visible in the field of view of the embedded camera (exterior red dotted line around the UAV in Fig. 6).
- For convenience we call 'collision' the situation where the UAV will fly over an object.
- A collision is detected if the direction of an object's motion vector intersects the path in front of the UAV.
- A danger area is computed, and is visible as the smallest red dotted rectangle in Fig. 6, for every frame depending on UAV's velocity so that the UAV will reach the boundary in 5 seconds at current and constant velocity.

$$\Delta p = \frac{V_{UAV}}{F_d \cdot r_m} \quad (8)$$

where Δp is the number of pixels to advance along the path, V_{UAV} is the velocity of the UAV, F_d is the framerate of the dataset, and r_m represents the ground sampling distance of the geo-registered map.

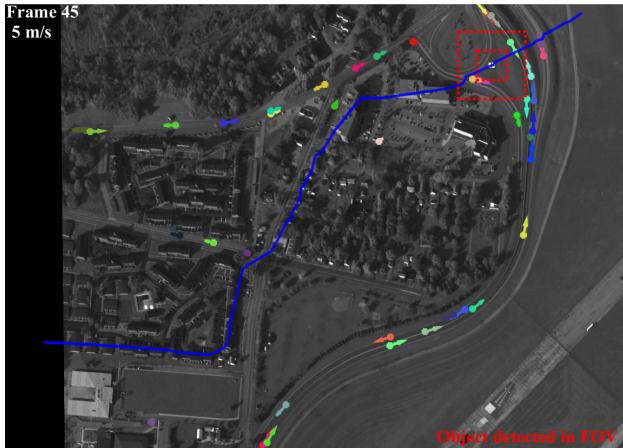


Fig. 6. Small red dotted square on the top right represents the danger area that the UAV would reach in 5 seconds at the current velocity, and the larger square shows the FOV. Objects locations and their motion vectors given by ground truth are shown by colored arrows. At bottom right a notification is displayed in red if objects are present in the FOV.

B. Datasets

We use two datasets to run our safe navigation pipeline, Wright-Patterson Air Force Base (WPAFB) [1] and PVLabs. They are wide-area motion imagery (WAMI), and provide ground truth for moving objects on ortho-rectified images captured by UAVs. Both of those datasets have been captured at high altitude with embedded sensors and a matrix of multiple cameras. We use the provided regions of interest outputted by a geo-registration step, described in [1]. For each dataset we run the different steps of the pipeline. We first create the weight map using the process described in

section III-A. Videos are then precisely geo-registered onto the map via homography transformation. The global path (Fig. 7) is generated before the simulated flight and adapted dynamically on the way.

V. RESULTS

For both WPAFB and PVLabs datasets, we defined 9 different pairs of start and end GPS coordinates (Fig. 7) based on the environment and busyness of the roads to create challenging situations that will require global path adaptation. And each path is executed at three different UAV velocities: 5, 8, and 11 m/s. The total traveled distance by using the global path compared to the classic straight line path, for each dataset executed for all paths at three above velocities, is 20% higher or 5:13 min longer for WPAFB and 6% or 32 s for PVLabs, making our safety increased path an affordable measure in term of autonomy.

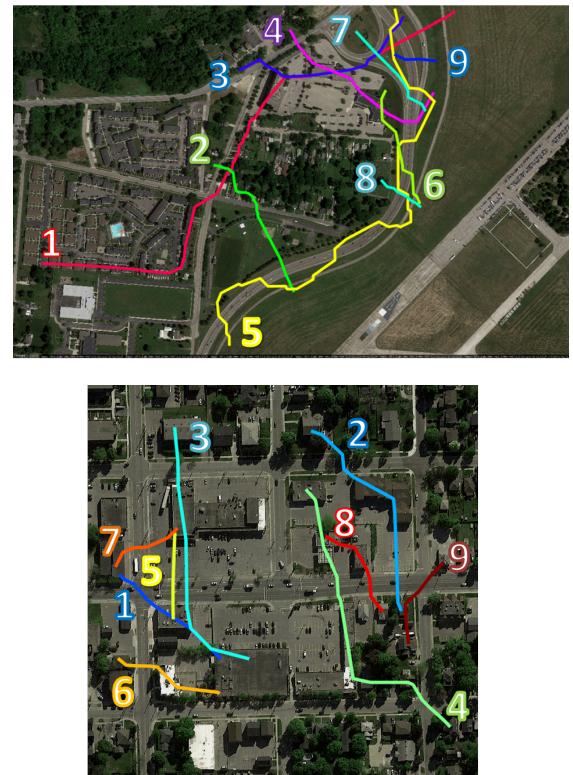


Fig. 7. Visualization of the nine paths that have been tested (each at 5, 8, and 11 m/s) for both datasets. Top: WPAFB. Bottom: PVLabs. Images acquired using the Google Maps API.

To quantify the performance of the proposed method we introduce a metric assimilated to safety. We consider the UAV to object proximity, the closer the UAV is to an object the more danger it represents for it, we therefore compute a total cost for each dataset as in (9).

$$C_g = \sum_{i=1}^{\#Objects} \alpha \cdot e^{-D_{ou}^i} \quad (9)$$

with C_g the global cost for the considered dataset, D_{ou} the ground distance between the UAV and each object

detected in the FOV during the experiment, and α a constant.

Note that, for the same start and end locations, when different paths are compared, the UAV will not encounter the same situations. This is why, for clarity, we include, with the results in Table I, the number of objects seen by the UAV's camera throughout the simulation for each dataset.

TABLE I
SAFETY ESTIMATION RESULTS FOR WPAFB AND PVLABS

	Straight path	Static path	Dynamic path
WPAFB # of obj.	2,759	4588	7,597
Global WPAFB cost	243.9	62.9	5.6
PVLabs # of obj.	3,600	4,022	5,959
Global PVLabs cost	188.1	326.3	98

We can clearly see in Table I that our proposed method encounters more objects in the FOV, but it has the means to keep the UAV afar from them. Objects which are over 20m away are not in danger, but having a car or pedestrian closer than 5m to the UAV represents a very concerning situation in terms of safety for people. This is why we have chosen to compute the global cost with a negative exponential weight function, that way the shorter the distance, the more cost is applied to the global metric. The proposed method encounters over twice the amount of moving objects but safely keeps away from them (Fig. 8), making the resulting safety parameter much better than global path and, most of all, better than classic straight line path.

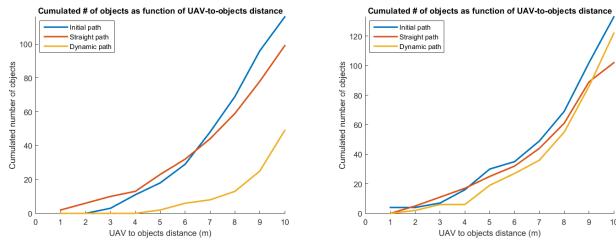


Fig. 8. Comparison of the number of detected objects in the FOV as function of UAV-to-objects ground distance between 0 and 10m for all nine paths executed at 5, 8, and 11 m/s. The perfect solution would be 0 objects for all distances. Left: WPAFB. Right: PVLabs.

VI. CONCLUSION

In this paper we introduced an environment and safety based path planning for low altitude UAV operating in urban areas. We compute a global path, for any mission given a pair of start and end GPS locations, by using a weighted shortest path. The weight map is defined using ground classification data summarized in three classes: highest cost is for roads and paths because of the high probability of presence of people for which the UAV represents a safety threat, safest are buildings and water, and neutral areas are the rest. Additionally, we included a dynamic path planning that will modify locally the flight plan while in flight to avoid being

close to moving objects such as vehicles and pedestrians. Our proposed method has been tested in simulation using geo-registered data and images from two WAMI datasets, WPAFB and PVLabs, and it showed significant improvement compared to the current and manual mission planning solution in terms of a safety metric quantifying threat in function of UAV-to-object distance.

Our safety planning and navigation scheme can be implemented on-board a UAV and will consist in the following steps: 1- before takeoff, acquire necessary GIS data for the mission area, and generate mission waypoints using global weighted path planning, 2- during the flight, geo-register the embedded camera's images using a sensor model and gimbal readings, detect moving objects (as in [3]) or any other type of objects to avoid, and generate new local path and waypoints to stay clear of the detected objects.

ACKNOWLEDGMENT

The research was supported by a DGA-MRIS scholarship.

REFERENCES

- [1] Cohenour et al., "Camera models for the wright patterson air force base 2009," IEEE Aerospace and Electronic Systems Magazine, 2015.
- [2] Sheikh et al., "Geodetic Alignment of Aerial Video Frames," in Video Registration, Eds. Boston, 2003.
- [3] Castelli et al., "Moving object detection for unconstrained low-altitude aerial videos, a pose-independant detector based on Artificial Flow," ISPA, 2015.
- [4] Quigley et al., "Target Acquisition, Localization, and Surveillance Using a Fixed-Wing Mini-UAV and Gimbaled Camera," ICRA, 2005.
- [5] Kimura et al., "Automatic extraction of moving objects from UAV-borne monocular images using multi-view geometric constraints," IMAV, 2014.
- [6] Teutsch et al., "Evaluation of object segmentation to improve moving vehicle detection in aerial videos," AVSS, 2014.
- [7] Xiao et al., "Vehicle detection and tracking in wide field-of-view aerial video," CVPR, 2010.
- [8] Lin et al., "Efficient detection and tracking of moving objects in geo-coordinates," Machine Vision and Applications, 2011.
- [9] Rafi et al., "Autonomous target following by unmanned aerial vehicles," SPIE 6230, 2006.
- [10] van Toll et al., "Dynamically Pruned A* for re-planning in navigation meshes," IROS, 2015.
- [11] Xu et al., "Real-time 3D navigation for autonomous vision-guided MAVs," IROS, 2015.
- [12] Dumble et al., "Airborne Vision-Aided Navigation Using Road Intersection Features," Journal of Intelligent & Robotic Systems, 2015.
- [13] Pritt et al., "Georegistration of multiple-camera wide area motion imagery," IGARSS, 2012.
- [14] Habbecke et al., "Automatic registration of oblique aerial images with cadastral maps," Trends and Topics in Computer Vision, 2010.
- [15] Tchernykh, "Optical flow navigation for an outdoor UAV using a wide angle mono camera and DEM matching," IFAC, 2006.
- [16] Hrabar et al., "Combined optic-flow and stereo-based navigation of urban canyons for a UAV," IROS, 2005.
- [17] Achtelik et al., "Onboard IMU and monocular vision based control for MAVs in unknown in-and outdoor environments," ICRA, 2011.
- [18] Pestana et al., "Computer vision based general object following for GPS-denied multirotor unmanned vehicles," ACC, 2014.
- [19] Pestana et al., "Vision based gps-denied object tracking and following for unmanned aerial vehicles," SSRR, 2013.
- [20] Ess et al., "Object detection and tracking for autonomous navigation in dynamic environments," International Journal of Robotics, 2010.
- [21] Israelsen et al., "Automatic collision avoidance for manually tele-operated unmanned aerial vehicles," ICRA, 2014.
- [22] Gonzalez et al., "Using state dominance for path planning in dynamic environments with moving obstacles," ICRA, 2012.