2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Learning to Predict Obstacle Aerodynamics from Depth Images for Micro Air Vehicles

John Bartholomew, Andrew Calway and Walterio Mayol-Cuevas
{jpab,andrew,wmayol}@cs.bris.ac.uk
Department of Computer Science
University of Bristol, UK
Bristol Robotics Laboratory, UK

***Abstract*— Many applications of Micro Air Vehicles (MAVs) require them to operate in cluttered environments, flying in constrained spaces and close to obstacles. Such obstacles affect the airflow around the MAV and can thereby affect its flight characteristics. We describe a system for predicting these effects at a distance, using depth images obtained from an RGB-D sensor. Predictions are based on learning from prior experience gathered during training flights. We show that aerodynamic effects caused by obstacles are consistent, and demonstrate that it is practical to make predictions from experience without running a computationally expensive aerodynamic simulation. Our approach uses a Gaussian process regression, it requires minimal parameter tuning and is able to predict the acceleration that will be expected at a distance in the future. The method produces estimates within 12ms without any code optimisation and the results indicate good prediction ability with mean errors within 4-10cm/s^2 on a database of various obstacles.**

I. Introduction

Micro Air Vehicles (MAVs) have several potential applications that present a requirement to operate within cluttered and tightly constrained environments. Any application for which the MAV is expected to operate indoors is likely to have these characteristics. In such an application, a MAV will frequently be flying in close proximity to the surfaces that make up its environment, and it will be subject to aerodynamic interactions with those surfaces.

These aerodynamic interactions are most commonly ignored in control and path planning, or treated as random disturbances which the controller must reject through feedback. This random disturbance model is perhaps necessary if the MAV's flight environment is unknown, but in fact for a lot of practical applications MAVs must carry sensors that provide information about the shape of the MAV's surroundings.

In robotics in particular, physical interactions with the environment are essential, but very difficult to model. In some cases they can be simulated, but such simulations (*e.g.*, computational fluid dynamics) may be too computationally expensive to be incorporated into a control loop or real-time path planning system. Prediction based on past experience is an alternative to simulation which avoids this and other difficulties such as accurate platform modelling and the production of adequately realistic input data.

Being able to predict aerodynamic effects has multiple potential uses for MAVs. At the path planning level, predictions could be used to determine a more precise safety margin through confined spaces, or to select paths that are likely to suffer the least disturbances. Perhaps more interesting, a path planner could deliberately select paths on which predicted disturbances provide some of the impulse required to follow the path, thereby reducing the total energy requirements. At the controller level, predictions of aerodynamic effects could be incorporated in a feed-forward term in the control loop, allowing the desired trajectory to be followed more accurately.

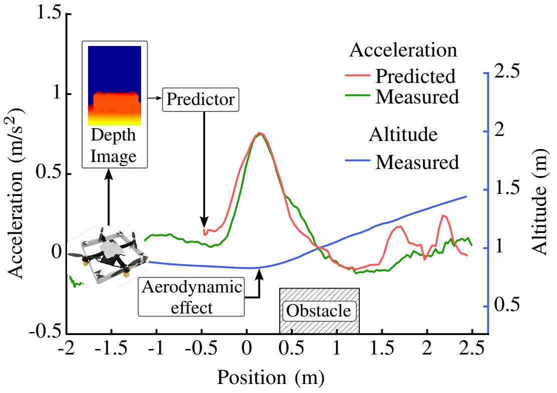

Fig. 1: Our framework learns to predict the acceleration due to aerodynamic interactions between a MAV and nearby obstacles. Using depth images as input, the system predicts the acceleration that will be experienced at a fixed distance in the future (here, 1.5 m ahead). As the platform moves toward the obstacle, the acceleration experienced will change and can become substantial as shown in the green plot (actual acceleration). The red plot shows our prediction result which closely follows the actual acceleration. Thrust is kept constant through the flight and is set to enable stable hover at the start. The flight path, in blue, shows that the effect of the interaction with the object is a clear deviation from level flight. See accompanying video for more results.

As we show in this paper, the effect that aerodynamic interactions have on a MAV's flight is largely consistent, and we develop a framework to predict these effects based on past experience. Our starting point is the framework we proposed in our previous work [1] for the prediction of landing behaviour on different materials using images, but here we modify it to use depth measurements for the prediction of aerodynamic effects from nearby obstacles.

978-1-4799-3684-7/14/$31.00 ©2014 IEEE

The paper is organised as follows. In section II, we discuss previous uses of learning for MAV control and modelling, and motivate our own research in this area. In section III we describe our data collection methodology and the hardware we use. In section IV, we give flight results to demonstrate the consistency of aerodynamic interactions with obstacles, and give some qualitative descriptions of our observations. In section V, we define and explain our learning and prediction system. In section VI, we discuss the effects of the tuning parameters in the predictor, and characterise its capabilities and effectiveness. Finally, in section VII we conclude by summarising our approach and its motivation.

II. Related Work

Most of the work concerned with learning methods for Unmanned Aerial Vehicles (UAVs) and MAVs has concentrated on tuning of the controller or modelling of dynamics when operating in free space, where aerodynamic interactions with nearby surfaces can be neglected.

Some examples of learning for flying platforms include the work of Ng *et al.* who learn a system model for an unmanned helicopter from data collected through manual flight, and then use reinforcement learning in simulation to design a controller capable of sustained inverted flight [2]. Abbeel *et al.* improve the model learning step [3], and extend the capabilities of the learned controller to a wider set of aerobatic manoeuvres [4]. More recently, Lupashin *et al.* have used an iterative learning method to tune parameters for a highly aggressive multi-flip manoeuvre on a quadcopter [5]. Mueller *et al.* use learning to correct for the effects of consistent but hard to model effects in quadcopter dynamics to improve the precision of a trajectory following controller [6].

Probably the most well-known aerodynamic interaction between rotary-wing platforms and nearby objects is 'ground effect', which results in increased lift when operating close to the ground. This can be disruptive but can also be exploited to improve lift performance [7]. In the MAV domain, Nonaka *et al.* developed a ground-effect-aware altitude controller for a small coaxial configuration R/C helicopter [8]. They empirically propose a second order polynomial function model of ground effect, conceding the difficulty of obtaining an analytic formulation. They perform characterisation and experiments of this effect in the neighbourhood of 0 mm to 400 mm from a flat surface. Ground effect is similarly measured in [9], which also examines 'ceiling effect' and notes the effect of other nearby quadrotors. Being aware of ground effects can in turn be used to improve the controller [8] and even consider ground effects resulting from the case of wind gusts as considered in simulation in [10]. In [11] a vision-based system aware of ground effects is used for landing a MAV.

Ground effect is but one of a number of aerodynamic interactions that a MAV operating close to obstacles may encounter and this combined with the difficulties of modelling the platform itself, has resulted in little work on characterising or compensating for this variety of aerodynamic interactions.

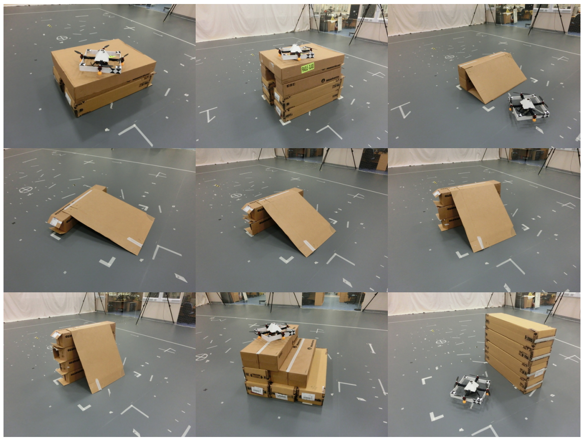

Fig. 2: Obstacle configurations. Shown: Boxes of different heights, ramps at different angles, steps, and a simulated wall.

In this paper we concentrate on the characterisation of the effect on multiple shaped objects and the parameters required for its prediction, and leave the use of this information within the control loop for future work.

As far as we are aware, this work is the first to attempt to both characterise aerodynamic interactions between a MAV and a variety of obstacles beyond a flat surface, as well as being able to learn and predict these effects from on-board visual data at a distance and in real-time.

Following we explain our experimental setup before proceeding with the description of the prediction framework and results.

III. Experimental Setup

As discussed above, one of the main objectives of this work is to learn complex aerodynamic effects from experience. This enables the learning of a variety of conditions which are not required to be explicitly formulated. This paradigm can be a useful competence for a class of MAVs intended to "go out and explore" with few assumptions and with an incomplete modelling of the world.

In this case, training and testing data for the predictor is gathered by performing a series of test flights over a variety of different obstacle configurations and observing the deviation from the expected flight path. We also collect depth images of the different obstacles with an RGB-D camera, as will be discussed below.

Most of the obstacle configurations are shown in Fig. 2, some others are made by re-positioning obstacles relative to the flight path e.g. approaching a ramp at 90° instead of front facing. The no-obstacle case was also considered.

The MAV is flown over the test obstacle several times to measure the effect that the obstacle has on the MAV's flight.

Flights were performed in our flying arena which is a $12 \times 15 \times 4$ metre space, instrumented with a 10-camera Vicon motion capture system. The flight platform is an off-the-shelf DJI 'Flamewheels' F450 quadcopter. An on-board

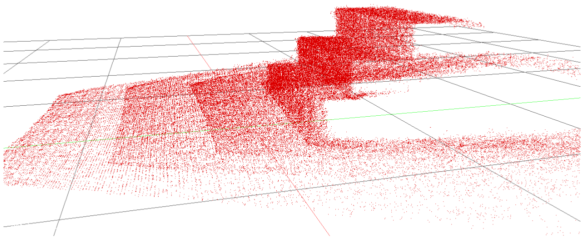

Fig. 3: The aligned point cloud for the ‘Steps’ obstacle.

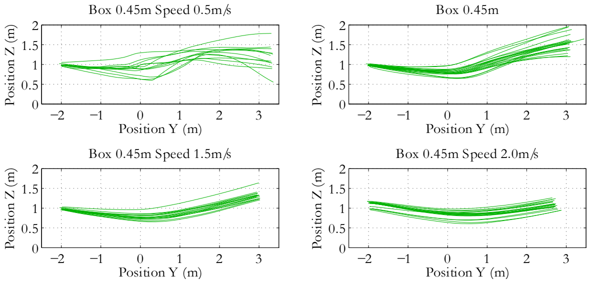

Fig. 5: The effect of varying flight speed.

DJI ‘Naza’ autopilot module is used in rate-control mode, and a R/C transmitter system is used to send throttle setting and roll-, pitch- and yaw-rate values to the platform.

A control system runs on a laptop computer to provide attitude and position control based on position and orientation feedback from the motion capture system. This control system runs at 50 Hz and consists of two layered control systems: A Proportional Integral Derivative (PID) controller for position, outputting attitude demand and throttle settings, and a proportional controller taking the demanded attitude and outputting roll-, pitch- and yaw-rate.

The experimental procedure is to fly a straight line path over an obstacle, moving the position set-point at a constant velocity. In order to simplify the behaviour of the system, the throttle setting is fixed while following this path. This means that the vehicle’s total thrust is (close to) constant, which isolates the effect of the obstacle. Prior to performing a pass over the obstacle, the position set-point is held at the start point (with throttle controller enabled) until the vehicle’s position has stabilised.

Along with flight data captured during these test flights, a structured light depth sensor (an ASUS Xtion Pro LIVE) is used to record each obstacle configuration. This provides sensor data with realistic characteristics, and this sort of sensor has been successfully used on board MAVs in other research. Due to limited payload capacity and other practical considerations in carrying this sensor on-board our platform, we collect these depth images with the sensor on a tripod. Several images are captured with the sensor positioned at intervals along the flight path, at the flying height of the vehicle. These depth images are then converted into point clouds, which are aligned and registered with the global coordinate system used in the flight area. Fig. 3 shows the point cloud for the ‘steps’ obstacle. Sensor images can then be synthesised with a virtual sensor following the exact flight path measured by the motion capture system.

Note that separating the flight paths from the 3D data representing the obstacles does not affect the validity of the framework or results presented, since the obstacles themselves are static during the flights; it is simply a practical consideration due to payload restrictions.

The resulting 3D obstacle data is used within the learning framework paired with corresponding flight paths for performing the aerodynamic effect predictions.

IV. Obstacle Effects

Fig. 4 shows the vehicle’s path for several flights over three of the obstacle configurations. The ‘No Obstacle’ case shows the variation in flight path caused by air movement in the room (*e.g.*, from open doors or ventilation systems) and differences in initial conditions (*e.g.*, slight error in the neutral throttle setting found during the pre-pass stabilisation period). The ‘Box 0.45m’ series shows the increase in thrust experienced when passing over a box of height 45 cm. This increase is quite consistent, though it varies somewhat depending on the height of the vehicle as it approaches the box. The ‘Ramp (Forward)’ series also shows a reasonably consistent effect, but the effect is clearly different from both the ‘No Obstacle’ case and the ‘Box’ case.

Aerodynamic effects can be less significant in certain circumstances due to inertia. Fig. 5 shows how the effect of the obstacle on the MAV’s flight varies as the speed of the MAV is changed. At the highest speed tested, with the position set-point moving at 2 m/s, the MAV is barely affected by the obstacle at all. At the lowest speed tested, the MAV is affected most strongly and its flight becomes more erratic.

We note that not every MAV mission can be flown at a speed that will reduce these aerodynamic effects, in particular for complex agile motions within tightly constrained environments, and we argue that it is not desirable for a MAV to ignore these effects altogether. The ability to predict the effect at different speeds could be very useful on its own and could form part of a more complete decision process. The vehicle could for example decide to fly at a speed known to minimise effects over an obstacle for which these effects are predicted to occur and then return to a more energy efficient exploration speed when the prediction recommends so.

V. Predicting Aerodynamic Interactions

Our main goal in this work is to make a useful prediction of the effect of aerodynamic interactions with the MAV’s local environment. We characterise this effect in terms of the acceleration that will be produced when going over obstacles of different shapes.

During flight, the forces acting on the MAV are: gravity, thrust from each rotor, torque from each rotor, drag from motion through the air, and a force from environmental air movement (*e.g.*, gusts). The effect of aerodynamic interactions with local surfaces can be summarised as an extra

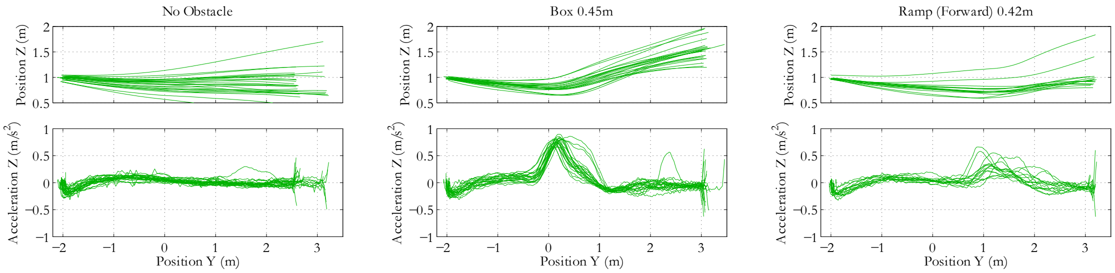

Fig. 4: Examples of flight path and acceleration over different obstacles.

force acting on the MAV. Since the MAV's mass is constant, this force can equally be characterised as the acceleration produced.

One of the advantages of our learning approach is that we do not have to explicitly model the various aerodynamic forces acting on the MAV, we just need to capture the net effect in the training data. In this work we do however assume that external wind gusts are negligible. We will come back to this aspect in future work.

In the general case, it is necessary to estimate the acceleration produced by the combination of other forces acting on the MAV, and subtract that from its net acceleration to determine the effect of its local environment. However, in our experiments we have held thrust constant at a value chosen to match the weight of the MAV. This means that when the MAV is level, the only vertical acceleration it experiences should be the result of environmental interactions, or global air movement.

As a simplification, we neglect the fact that the MAV is not always level, and the corresponding variation in vertical thrust. This effect is visible in the acceleration data as a negative acceleration at the start of the pass: here, the vehicle must accelerate forward to follow its set path, and this requires a pitch forward and corresponding reduction in vertical thrust (total thrust remaining constant due to the constant throttle setting).

In order to make this prediction, we must relate the measured acceleration during training flights with the geometry of the MAV's local environment during that flight. Formally, we have a set of n examples from training flights. Each example is a pair of some sensor data, $\mathbf{x}$ (in our case, a depth image from the RGB-D sensor), and some measurement, $\mathbf{y}$ (the acceleration experienced by the vehicle).

A. Gaussian process regression predictor

The basis of the predictor we use is the Nadaraya-Watson estimator [12], [13], which is a Gaussian process regression approach. Given a new input $\mathbf{x}'$, the output is estimated as a weighted average of the n example $\mathbf{y}$ vectors. The weight given to an example is determined by applying a kernel function (K_α; for which we use a Gaussian function) to the distance from the new input to the example.

$$\tilde{\mathbf{y}} = \frac{\sum_{i=1}^{n} \mathbf{y}_i K_\alpha(D(\mathbf{x}', \mathbf{x}_i))}{\sum_{i=1}^{n} K_\alpha(D(\mathbf{x}', \mathbf{x}_i))} \tag{1}$$

There are several choices that must be made in constructing this system: The form of the sensor data $\mathbf{x}$, the form of the output $\mathbf{y}$, the distance metric D, and the tuning parameter α, which is the bandwidth of the kernel function.

Each example pair consists of a depth image captured during a training flight, and a corresponding acceleration value, from some fixed distance ahead in the same flight (this distance is discussed in section VI-B). The task of the estimator is to compare the new input depth image to the training examples, to produce an estimate of the effect expected ahead of the current position.

B. Perceptual representation via depth images

For predicting aerodynamic interactions, we use the L^1 norm on the raw depth image data as the distance metric. While there have been various works on extracting features from depth images it seems sensible to first judge how well the system works when using the depth data directly. In particular there has been recent work on the direct use of depth data for relocalisation in 3D maps, which has shown that using small images is a feasible, robust and fast alternative to a feature-based approach [14].

Fig. 6 plots the distribution of image-to-image distances between the box-0.45m obstacle and all obstacles. This gives an indication of the discriminative power of this use of the L^1 norm, and the level of inter-class and intra-class variation that we observe.

We use depth images of 40×60 pixels (field of view $32°$ horizontal, $46°$ vertical), and as noted in section III, we synthesise our input images from a point cloud of the real scene. Depth values greater than 3 m are eliminated (set to zero). The limited field of view and depth range is because we are predicting the effect that the MAV will experience at a particular point in space, and we assume that the effect of obstacles is localised. We therefore only want to consider surfaces within a limited region around our target point.

The virtual sensor is positioned along the MAV's true flight path as reported by the motion capture data. The orientation of these virtual images is fixed globally, rather than being taken from the MAV's orientation. This is because the point for which a prediction should be made is in the direction of motion of the MAV, which is fixed in this experiment, whereas the MAV's orientation changes as necessary to follow the set path (*e.g.*, the MAV must pitch forward to accelerate forward).

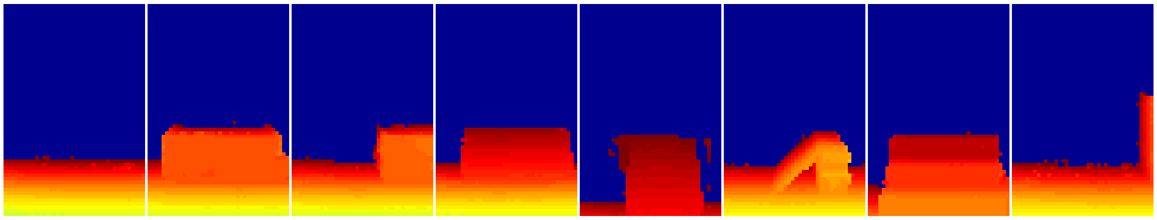

Fig. 7: Example depth images, after preprocessing to crop the edges and apply the depth threshold. The configurations shown are (from top left): no-obstacle; box-0.45m; edge-0.45m; ramp-forward-0.42m; ramp-forward-62deg; ramp-sideways-0.42m; steps-0.65m; flat-wall-0.88m.

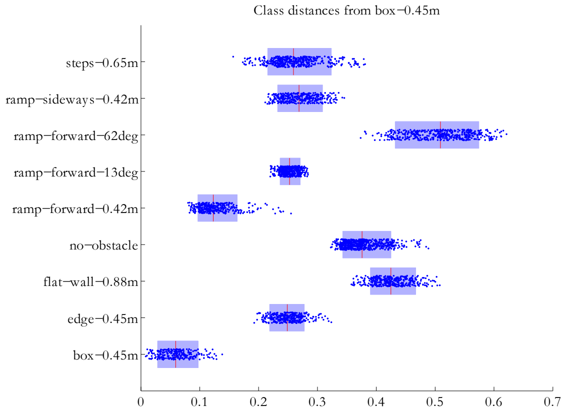

Fig. 6: Example of the distribution of class-to-class L^1 distances. These are the distances between images at a single time offset in the flight, so variation comes from differences in the MAV's viewpoint in different flights.

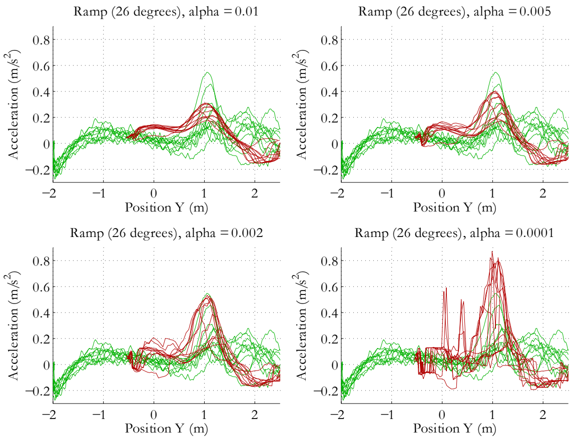

Fig. 8: The effect of varying the kernel function bandwidth. From left to right and top to bottom, α is: 0.01, 0.005, 0.002, 0.0001. Green: actual acceleration. Red: predicted acceleration.

VI. Prediction Results

In this section we present the results of our approach and highlight several aspects such as generalisation and size of prediction horizon, before showing the results with the entire database of obstacles Fig. 2.

A. Generalisation from learnt obstacles

In order to be useful in a wide context, the predictor must be able to produce sensible output when presented with objects that did not appear in the training set. That is, the predictor must be able to generalise from its training examples to other shapes that are "nearby" in some suitable space. The Nadaraya-Watson estimator should be capable of this, but its effectiveness depends on how the distance metric on the input vectors (depth images) corresponds with similar outputs (accelerations), and also on the bandwidth parameter of the kernel function. To test this, we ran flights over a series of ramps at different angles. We then trained the system on all but one of these ramp obstacles, and generated predictions for the flights over the untrained example. Fig. 8 shows the predicted and ground-truth acceleration for this test, with different kernel function bandwidths.

The selection of the kernel bandwidth involves various trade-offs. A large bandwidth will make the estimator more robust to noise, and more able to interpolate between training examples that are widely spaced in terms of depth image distance, but it also tends to reduce the discriminative ability of the estimator, eliminating true peaks in the training data. This can be seen in the first two plots in Fig. 8, where the prediction is smoother and has lower amplitude than the ground-truth. A small bandwidth will make the estimator give more accurate output when given an input that is close to a training example, but can lead to noise or drop-outs when given examples that are far from the training set. This noisy output is visible in the fourth plot in Fig. 8.

We selected a bandwidth of 0.002 as giving a good balance between discrimination and robustness to noise. Note that with this bandwidth, the predictor is able to interpolate to a useful degree between similar obstacles, to provide an estimate for acceleration over the "ramp-forward-26deg" obstacle, even though that obstacle was not included in the training set.

B. Size of prediction horizon

Another design consideration of the system is the prediction distance. This is the distance ahead of the MAV for which aerodynamic interactions are predicted. A large prediction distance is useful for path planning, but occlusions from nearer parts of the environment, or limitations of the sensor may make it harder to achieve. If the prediction is

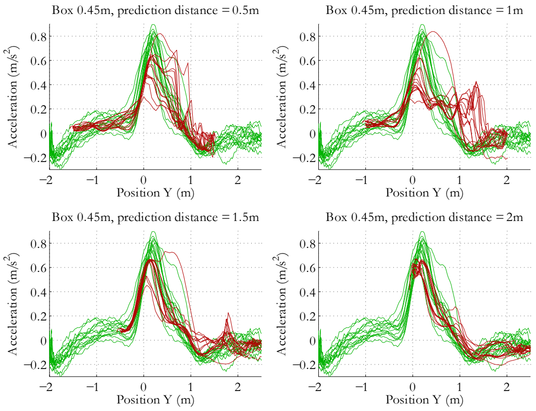

Fig. 9: The effect of varying the prediction distance. From left to right and top to bottom: 0.5 m, 1.0 m, 1.5 m and 2.0 m. Green: actual. Red: predicted.

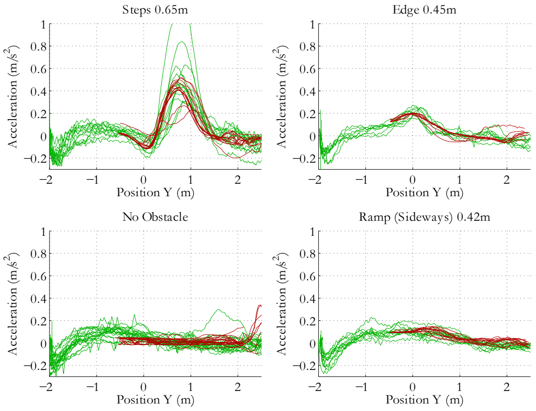

Fig. 10: Predictor output and ground truth, prediction distance 1.5 m, bandwidth 0.002, obstacles: steps-0.65m, edge-0.45m, no-obstacle and ramp-sideways-0.42m. Green: actual. Red: predicted.

to be fed into the controller, then a short prediction distance may be more appropriate.

Fig. 9 shows predictions for the "box-0.45m" obstacle at four different prediction distances. These results show a clear improvement in the quality of the predictions as the prediction distance is increased. This is most easily explained by the field of view and orientation of the (virtual) sensor from where synthetic depth images were obtained. In order to make good predictions, there must be a correlation between a depth image and the acceleration experienced in that region of space. With a forward-looking depth sensor, the space near to the vehicle is only visible in a narrow band at the bottom of the image. While the shape visible in this region of the image may correlate with acceleration only half a metre ahead, that correlation is obscured by the rest of the image, which correlates more strongly with the acceleration experienced at a greater distance.

C. Aerodynamic effects prediction results

To test our predictor, we use one third of our data for training and two thirds for testing. This split is done separately for each obstacle configuration by shuffling the flights for that obstacle, and selecting the first third of the result to be used as training data. Each obstacle has between 15 and 27 flights, with the mode being 16. For the majority of the tests, in which the set-point moves at 1 m/s, there are 275 motion capture samples. Of these samples, every second sample out of the first 60% is used to generate a synthesised depth image. We stop at 60% of the flight path samples because by that point the platform has passed over the obstacle and can no longer see it.

We use two test sets, which are different subsets of our data. Our general test set contains the full variety of obstacles for which we ran test flights, and our ramp series test set, which contains ramps at six angles, from 0° (no obstacle) to 90° (a box). For this data series, as noted in VI-A, we exclude one of the six from the training data.

The general test uses a training set containing 4426 depth-image/acceleration examples. The ramp series test uses a training set containing 3015 depth-image/acceleration examples. Our predictor (implemented in MATLAB® with a native code implementation of the L^1 metric) takes approximately 12 ms on a 3 GHz Intel Core 2 processor to make a prediction on the general (4426-image) training set.

For a general test of the system, and based on the above experiments, we used a bandwidth of 0.002 and a prediction distance of 1.5 metres. Although the prediction is arguably slightly better at 2 metres, this is quite close to the distance at which the MAV starts, so it cuts off the start of the aerodynamic interactions. A distance of 1.5 metres was chosen as a trade-off between quality and coverage of prediction. Our general test set contains 9 obstacle configurations. We show results for four of these in Fig. 10 (other configurations include the box and ramps, for which results have been discussed already). Even though the effects themselves are fairly small for some of these obstacles, the predictor output matches reasonably closely in most cases with the ground-truth.

D. Prediction Error

We characterise the performance of the predictor numerically by taking the mean prediction error (the prediction error being the absolute difference between the predicted acceleration and the measured acceleration) for each flight, and then finding the mean and standard deviation across all test flights for a particular obstacle. Results are shown in table I.

Several obstacles are considered and for the general test cases include steps of 0.65m total height, ramps at various angles (13° and 62°), free space (no-obstacle), passing by a wall/column of height 0.88m, flying following the edge of a 0.45m high box as well as going forward over that same box. In the second data-set which is used mainly to showcase ability to generalise as shown in Fig. 8, has ramps of various angles including free space and a 0.88m box. As we can see from these results the error in the predicted acceleration is within 4 to 10 cm/s^2 and the error for the same obstacle on the different data-sets (which contain a difference of 1400

TABLE I: Predictor Average Error

General Series

Obstacle	Mean	Std.Dev.
steps-0.65m	0.076	0.040
ramp-sideways-0.42m	0.042	0.013
ramp-forward-0.42m	0.101	0.044
ramp-forward-62deg	0.055	0.016
ramp-forward-13deg	0.058	0.020
no-obstacle	0.044	0.019
flat-wall-0.88m	0.047	0.015
edge-0.45m	0.042	0.009
box-0.45m	0.103	0.040

Ramp Series

Obstacle	Mean	Std.Dev.
box-0.88m	0.107	0.046
ramp-forward-62deg	0.056	0.012
ramp-forward-41deg	0.079	0.015
ramp-forward-26deg	0.130	0.050
ramp-forward-13deg	0.057	0.019
no-obstacle	0.043	0.017

Prediction error (m/s^2). Mean and standard deviation of per-flight mean prediction error.

examples as explained before) is marginal in the order of a few mm. This last point is encouraging for scalability of the approach.

VII. Conclusions and Future Work

We have demonstrated that aerodynamic interactions between a MAV and its local environment produce consistent and measurable effects on its flight. As far as we are aware, this work is the first to attempt to both characterise these aerodynamic interactions for a variety of obstacles beyond a flat surface, as well as being able to learn and predict these effects in real-time.

The approach treats the problem as one of learning the relationship that exists between depth images of obstacles observed at a distance with the aerodynamic effects experienced by the platform when heading toward the object at a given speed. The method is based on a Gaussian process regression for which we have shown the effects of kernel bandwidth selection on its ability to generalise for similar objects not seen by the system as well as effects of predicting at different distances ahead.

The ability to predict these complex effects offers an alternative to aerodynamic simulation which is demanding in computation time, hardware requirements and energy consumption. In addition, prediction via experience retains the potential to cope with a wide variety of aerodynamic effects which would be difficult to capture for an analytic model.

Being able to predict what lift will be experienced ahead can be used for high level decisions: avoiding expected disturbances, or perhaps even intentionally passing over objects to gain lift. It can also be used in closed loop control, predicting ahead of time how to compensate for obstacle effects (in our experiments we have demonstrated predictions 1.5-2m ahead using a standard depth camera). This paper concentrates on characterising the effects and formulating the prediction framework while higher level tasks and integration with the controller is left to be demonstrated in future work.

Also note that we are not performing object detection, which has the benefit that we don't impose any arbitrary categories on the system. However it is conceivable that as more object classes are learnt and as the complexity of scenes considered increases, some form of object clustering or tree structured database could be useful to improve the robustness or scalability of the system. Scalability in particular is important to maintain real-time operation.

The approach we have developed is generic and can be easily adapted to accommodate learning of other effects beyond lift, for example predicting roll and pitch disturbances, or the effects of gusts when flying near objects.

Acknowledgements

This work was partially funded by the UK EPSRC. We thank Colin Greatwood for his efforts and help in using the Bristol Robotics Laboratory Flying Arena, and his advice on the practical aspects of quadrotor construction and flight.

References

[1] J. Bartholomew, A. Calway, and W. Mayol-Cuevas, "Predicting micro air vehicle landing behaviour from visual texture," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 4550–4556.

[2] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, "Autonomous inverted helicopter flight via reinforcement learning," in *ISER*, 2004, pp. 363–372.

[3] P. Abbeel, V. Ganapathi, and A. Ng, "Learning vehicular dynamics, with application to modeling helicopters," in *Advances in Neural Information Processing Systems 18*, 2006, pp. 1–8.

[4] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in Neural Information Processing Systems 19*, 2007, pp. 1–8.

[5] S. Lupashin, A. Schollig, M. Sherback, and R. D'Andrea, "A simple learning strategy for high-speed quadrocopter multi-flips," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 1642–1648.

[6] F. Mueller, A. Schoellig, and R. D'Andrea, "Iterative learning of feed-forward corrections for high-performance tracking," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012, pp. 3276–3281.

[7] J. Rayner, "On the aerodynamics of animal flight in ground effect," *Philosophical Transactions Royal Society London B*, vol. 334, no. 1269, pp. 119–128, 1991.

[8] K. Nonaka and H. Sugizaki, "Integral sliding mode altitude control for a small model helicopter with ground effect compensation," in *American Control Conference (ACC), 2011*, 2011, pp. 202–207.

[9] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. Kumar, "Influence of aerodynamics and proximity effects in quadrotor flight," in *Experimental Robotics*, 2013, vol. 88, pp. 289–302.

[10] T. Roy, M. Garratt, H. R. Pota, and M. Samal, "Robust altitude control for a small helicopter by considering the ground effect compensation," in *Intelligent Control and Automation (WCICA), 2012 10th World Congress on*, 2012, pp. 1796–1800.

[11] Z. Yu, K. Nonami, J. Shin, and D. Celestino, "3d vision based landing control of a small scale autonomous helicopter," *International Journal of Advanced Robotic Systems*, vol. 4, no. 1, 2007.

[12] E. Nadaraya, "On estimating regression," *Theory of Probability & Its Applications*, vol. 9, no. 1, pp. 141–142, 1964.

[13] G. S. Watson, "Smooth regression analysis," *Sankhy: The Indian Journal of Statistics, Series A (1961-2002)*, vol. 26, no. 4, pp. 359–372, 1964.

[14] A. P. Gee and W. Mayol-Cuevas, "6d relocalisation for rgbd cameras using synthetic view regression," in *Proceedings of the British Machine Vision Conference (BMVC)*, September 2012.