Received: 31 January 2017 | Revised: 18 May 2017 | Accepted: 3 July 2017

DOI: 10.1002/rob.21744

REGULAR ARTICLE

WILEY

R-ADVANCE: Rapid Adaptive Prediction for Vision-based Autonomous Navigation, Control, and Evasion

Hector D. Escobar-Alvarez[1] | Neil Johnson[1] | Tom Hebble[1] | Karl Klingebiel[2] | Steven A. P. Quintero[2] | Jacob Regenstein[1] | N. Andrew Browning[1]

[1]Scientific Systems Company Inc., Woburn, Massachusetts, 01801

[2]AeroVironment, Simi Valley, California, 93065

Correspondence
Hector D. Escobar-Alvarez, Active Perception and Cognitive Learning, Scientific Systems Company Inc., Woburn, MA 01801.
Email: hector.escobar@ssci.com

Abstract

In this article, we present a monocular visual reactive navigation system capable of navigating at high speeds, without GPS, in unknown complex cluttered environments. The system, called R-ADVANCE (Rapid Adaptive Prediction for Vision-based Autonomous Navigation, Control, and Evasion), consists of a set of biologically inspired visual perception and reactive control algorithms that provide low-computation reactive obstacle avoidance while navigating at high speeds in search of a goal object. These algorithms, along with basic planning, and augmented with low-precision visual odometry, were implemented on a micro unmanned aerial vehicle and tested in a number of challenging environments. While each of the individual algorithmic and hardware elements has been previously studied in limited environments, this work is the first time that these novel components have been integrated and flight-tested. To achieve fast flight, an NVIDIA Tegra TK1 was used as the main processor, allowing us to parallelize the system to process 1280 × 720 video streams at 40 fps, reaching flight speeds up to 19 m/s (≈68 km/h) or 42 mph.

KEYWORDS

aerial robotics, GPS-denied, High-speed, Obstacle avoidance, perception

1 | INTRODUCTION

Small and micro unmanned aerial vehicles (UAVs) have significant potential to change the way we live, from personalized package delivery that finds us where we are, to agricultural crop and pest monitoring. Currently, however, operational platforms are primarily utilized for autonomous GPS-based operation in limited, controlled environments or operation under constant human control. Most conceivable uses of these vehicles require fast autonomous flight in clutter with the ability to fly without operator control when radio link is lost and the ability to operate with periodic loss of high-precision GPS signals while flying in dynamic environments with potentially out-of-date maps. Autonomous systems need to be able to fly in sparsely cluttered natural environments, densely cluttered environments with combinations of man-made and natural obstacles, and potentially in highly dense man-made clutter, such as inside buildings.

To operate with loss of radio link to an operator ground (or air) control station, this type of autonomy system requires low size, weight, and power (low-SWP) onboard processing, with no unnecessary communications. Moreover, autonomy is an enabler of the overall system, but it is not the reason for using the system, and as such, the impact of the autonomy on the payload capacity of that aircraft should be further minimized. Currently, no such system exists, although much work is being performed on visual-, laser-, and ultrasound-based algorithms and compute/sensor packages.[1–4] In previous work using monocular cameras, speeds of up to 1.5 m/s were achieved on a quadrotor[5] using imitation learning techniques to simulate human pilot commands to avoid trees. Ground vehicles navigating in cluttered environments have achieved speeds of up to 4 m/s using a combination of machine learning, laser range-finders, and a learned function that predicts collision probabilities, but this technique has not been applied to flying vehicles.[6] A stereo vision system implementation on a field-programmable gate array (FPGA) was able to achieve flights between 1 and 5 m/s onboard a quadrotor;[7] still, this approach is not fast enough to enable higher speeds. An approach merging monocular and binocular vision enabled a system to reach speeds of up to 4 m/s but only in indoor environments, and it did not enable the autonomous detection and avoidance of obstacles.[8] In research using fixed-wing vehicles, speeds of up to 10 m/s were achieved, but knowledge of the environment was necessary to estimate the vehicle's position accurately using a laser range finder. [9] A very similar approach was used for indoor navigation of a quadrotor achieving speeds of up to 8 m/s, but again a known map of the environment was needed.[10] Flights in outdoor environments using a fixed-wing vehicle achieved speeds up to 9 m/s using

a pushbroom stereo vision algorithm.[11] The R-ADVANCE (Rapid Adaptive Prediction for Vision-based Autonomous Navigation, Control, and Evasion) implementation described herein has demonstrated the ability to navigate in indoor and outdoor environments at speeds ranging from 6 to 19 m/s. R-ADVANCE automatically scales speed to the level of clutter in the environment, and it is able to navigate in a range of environments, including but not limited to the following: unknown tree forests, urban outdoor building environments, indoor warehouse-like environments, and indoor to outdoor transitions through 5 × 5 m garage doors.

The goal of this research is develop a robust low-SWP flight system that can navigate quickly without GPS in complex unknown cluttered environments. R-ADVANCE addresses this problem by using a set of biologically inspired visual perception and reactive control algorithms, together with a basic planner, that provide low-SWP reactive obstacle avoidance while navigating in unknown environments at high speeds in search of a goal object. R-ADVANCE relies on a monocular camera for obstacle avoidance and goal detection. The core perception algorithm is based on the concept of Expansion Rate (ER), which is the inverse of time to contact (TTC).[12–14] High or low magnitudes of expansion rate indicate a respective high or low risk of collision. This obstacle representation can be easily incorporated into a potential field that is the link between visual perception and vehicle control, i.e., the steering field.[13,15,16] The steering field generates trajectories through instantaneous turning decisions to steer around obstacles toward a defined goal object. To meet the fast and precise image-processing requirements needed to achieve reliable performance at higher speeds, we use the NVIDIA Tegra TK1 processor. Through concurrent central processing unit (CPU) and graphics processing unit (GPU) processing, we are able to parallelize our processes, thereby increasing the overall processing rate of our system. With the current implementation, R-ADVANCE runs at around 40 fps. R-ADVANCE uses the GPU for the following:

- Image enhancement (adaptive histogram equalization).
- Image pyramids—Optical flow (VisionWorks Lucas-Kanade).
- Harris corner detector.
- Image data handling.

R-ADVANCE uses the CPU for the following:

- Target recognition.
- Customized adaptive exposure control.
- Expansion rate computation.
- Steering field potential field controller.

The system was implemented on a DJI Flamewheel F450 quadrotor, with E600 motors, 12-in. propellers, and a COTS 3DR Pixhawk autopilot. R-ADVANCE has successfully navigated and avoided obstacles inside unknown indoor and outdoor environments achieving speeds up to 19 m/s over distances up to 500 m. The full vehicle, including the R-ADVANCE payload and multiple debug radios, weighs around 2 kg. The R-ADVANCE payload has a clear path to a sub-200g and sub-15W implementation. There are smaller, lighter, lower-power systems in existence,[17–19] but they have not been demonstrated in a wide range of environments or at high speeds.

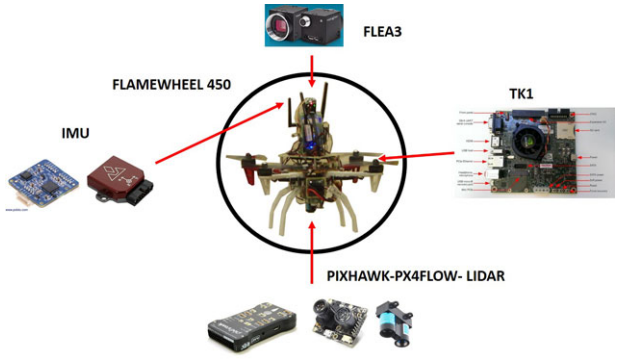

FIGURE 1 Hardware integration

2 | HARDWARE INTEGRATION

The hardware selected for the system provides a path to a deployable low-SWP implementation using easily available commercial-off-the-shelf (COTS) components. The hardware consists of three groups of components, namely the core perception sensors, the primary embedded computer processor, and the low-level navigation components.

The main processor chosen for interfacing with the perception sensors, processing the sensor data, and integrating the algorithms within a fully functional system architecture is the NVIDIA Jetson TK1 developer kit,[20] shown in Figure 1. This embedded platform has a GPU with 192 CUDA cores and a quad core ARM® CPU. This processor was chosen due to its ability to handle and process 1280 × 720 images at rates of up to 70 fps in a form factor that supports significant miniaturization in future research. The R-ADVANCE perception algorithms are well-suited to take advantage of the dual processing architecture with intensive imaging operations implemented in parallel on the GPU and inherently serial operations implemented on the CPU. At present, we are utilizing the GPU with a load of 25%, and the CPU with a load of 30% during fast autonomous flight without goal search on. The goal search was highly optimized to use all the remaining resources available on the CPU, and therefore when it is on, 90% of the processing power is consumed.

The core perception sensors of R-ADVANCE consist of a USB3.0 monocular camera with an S-mount lens, an inertial measurement unit (IMU), and a forward-facing single-point laser rangefinder. The selected perception camera is a color Flea3 camera[21] model FL3-U3-20E4C-C, with global shutter and an 8 mm lens that provides a 120° field of view. The camera is mounted facing forward, angled upwards 15° to account for vehicle pitch during forward flight, and it is configured to stream YUV images with a resolution of 1280 × 720 at up to 70 fps. Expansion rate and goal detection both use the imagery from the forward-looking Flea3 sensor. The selected IMU is a VectorNav VN-100[22] that is mounted on the camera to provide camera rotation information at rates of up to 200 Hz. We had previously integrated a CHRrobotics IMU,[23] but due to low accuracy and high drift,

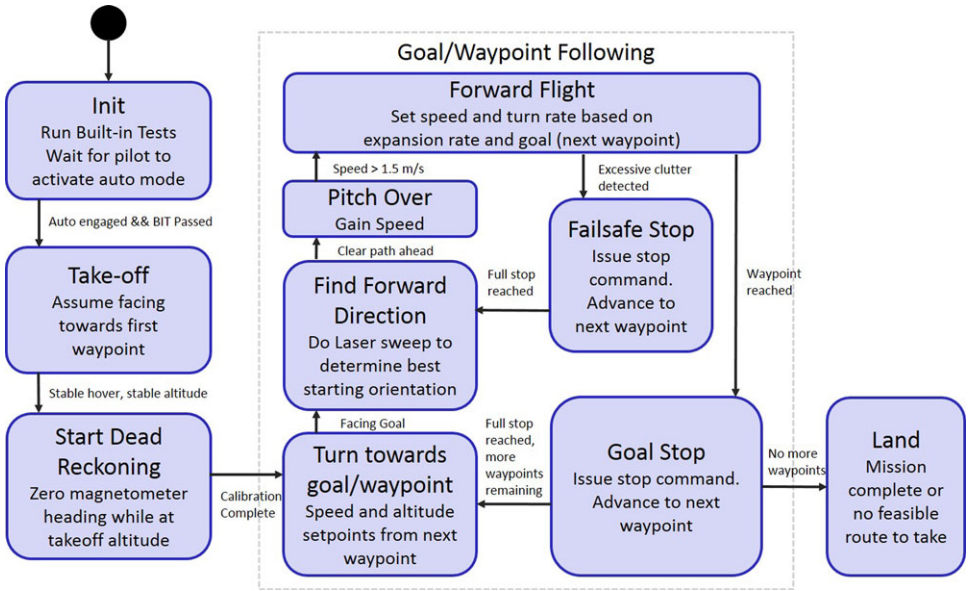

FIGURE 2 Executive planner state machine

the IMU was changed to the VectorNav. The forward laser rangefinder is a LiDAR-Lite V2.[24] The single-point laser rangefinder is used when the vehicle is stopped to provide a quick scan via vehicle yaw to determine which direction is safest to travel. It is not used during forward flight. The camera, lens, IMU, forward laser, and associated mounting hardware and cabling combined weigh less than 100 g (including all standard casings) and draw less than 5 W.

The navigation components of the system consist of a 3DR Pixhawk autopilot[25] combined with a 3DR PX4Flow optic flow sensor, a pitot-static airspeed sensor, and a downward-pointing LiDAR-Lite V2 point source laser altimeter. The PX4Flow and LiDAR enable velocity estimation at low altitudes, without GPS, at speeds up to 10 m/s. The airspeed sensor enables velocity estimation at higher speeds and higher altitudes. Low-level controls are handled by the PixHawk autopilot, which utilizes the open-source ArduCopter software suite with modifications for custom altitude control, speed control, and position estimation. A new flight mode enabling forward flight with coordinated turns was also added to ensure that the rigidly mounted Flea3 camera and pitch-compensated airspeed sensor face the direction of travel. This mode is used in addition to the standard position hold, auto-takeoff, and auto-land flight modes. The autopilot, PX4Flow, downward laser, airspeed sensor, and associated mounting hardware and cabling combined weigh less than 100 g and draw less than 2 W. The TK1 communicates with the PixHawk via custom Mavlink messages over a serial connection to command yaw rate, altitude, and speed.

3 | SOFTWARE ARCHITECTURE

The R-ADVANCE system software consists of three core components: the executive planner, the visuo-reactive navigation system, and the goal detection and recognition system. Each of these subsystems is described in detail below.

3.1 | Executive planner

The executive planner is implemented as a state machine that oversees the behavior of the vehicle while executing a mission. A mission consists of traversing to an end location defined by the presence of a goal object and with an expected position defined by (x_f, y_f) coordinates. The mission can optionally incorporate multiple intermediate waypoints defined either by relative position or by the presence of a waypoint object. The mission starts from a known initial location (x_0, y_0). The system must avoid obstacles as it traverses toward the goal object. The executive planner controls all transitions between vehicle states based on mission status and issues commands to the autopilot for takeoff/landing, visual goal/waypoint updates, forward flight reactive navigation mode, and yaw search for preferred direction of travel. The executive planner maintains a position filter that incorporates visual odometry data, from the PX4Flow and autopilot position filters, with visual goal detections to estimate the bearing and range to the goal object. The executive planner additionally handles communications with the perception sensors (IMU, LIDAR) and ensures that the information is current and available for the processes executing the perception and goal detection algorithms. The state machine can be seen in Figure 2. The executive planner executes in its own process running at 50 Hz.

3.2 | Image processing

The visuo-reactive navigation system, based on ViSTARS,[13] is comprised of a number of core algorithms: dynamic camera exposure control, IMU-based rotation correction, feature (contrast) enhancement, optic flow estimation, focus of expansion (FOE) estimation, ER computation, and steering field computation. The pixel intensive image processing functions are implemented on the GPU through the use of the NVIDIA VisionWorks SDK, while the FOE, ER, and steering field

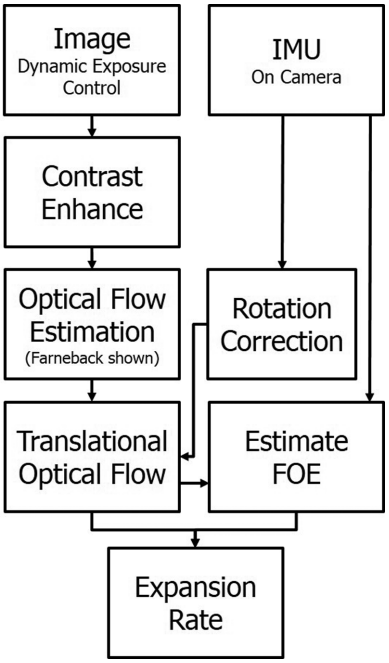

FIGURE 3 Flow diagram of image processing and expansion rate generation

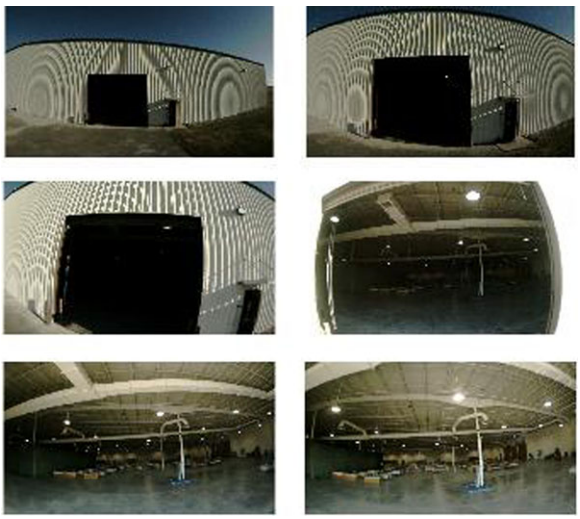

FIGURE 4 Autoexposure control: transition from outdoor environment to indoor environment shows automatic exposure control and clarity on the images. Every 5th frame is shown

controller are implemented on the ARM processor. A flow diagram of this process through to the ER can be seen in Figure 3.

Autoexposure was implemented using the algorithm from Ref. 26; a basic result is shown in Figure 4. The algorithm adjusts the camera exposure and gain in order to maximize the gradient content in the scene. Then the image is passed through a contrast limited adaptive histogram equalization (CLAHE) algorithm that enhances regions of the image that are either too dark or too bright to provide more reliable features. The rotation correction of the image is done as in Ref. 27. Features are selected using the Harris corner algorithm[28] using VisionWorks.[29] For each image frame, a pyramid of the image is generated and compared with the rotation-corrected previous frame, using the Lukas-Kanade optical flow algorithm, to generate a translation motion vector representation.

Once the translation motion vectors are extracted, a two-dimensional (2D) expansion rate image is generated. Expansion rate is the inverse of time to contact (TTC), and it provides a behaviorally relevant measure of collision risk to obstacles in the environment.[13] Mathematically, ER can be defined as the rate of growth of an object per unit of time.[12] Defining x as the distance from the FOE in the image space, the expansion rate can be defined as the motion vector ratio shown in Eq. (1). The FOE is estimated as per Ref. 13,

$$\mathrm{ER} = \frac{1}{\mathrm{TTC}} = \frac{\dot{x}}{x} \tag{1}$$

For more details, see Ref. 13. Under certain assumptions, ER is equivalent to the mathematical quantity of divergence,[30] and it is related to the behavioral concept of looming, both of which have been previously used for robotic control.[31–33]

Expansion rate is a robust, low-computation, direct measure of collision risk across the visual field. It becomes more observable at higher speeds (scalable) and enables long-range sensing via visible light. The process for computing ER can be seen in Figure 5. From the 2D expansion rate image, the region of interest (ROI) of the image, corresponding to the area of space directly on the current flight line, is selected and sorted by columns. Then the row with the 75th percentile is selected to generate the 1D expansion rate vector. This 1D vector is the obstacle representation along the azimuth angle of the vehicle. The higher the magnitude, the closer an obstacle is to the vehicle in that specific azimuth angle.

This 1D ER vector is incorporated in the steering field to generate heading rate and can be used to generate clutter-dependent speed control. The steering field reactively adjusts the vehicle actuation (direction) to avoid obstacles, generate reasonable (humanlike[34]) vehicle trajectories, and fly to a goal. Obstacle perception from vision systems is naturally represented as a distribution in the direction space. Angular regions with obstacles are to be avoided (repulsive), while angular regions in the direction of the goal are to be sought (attractive). Therefore, we combine the 1D ER (obstacle) field with a goal field (derived from the bearing and range estimates from the executive planner) into the steering field, shown in Figure 6 to generate heading rate control outputs.

Note that the minimum detectable obstacle is a function of resolution, frame-rate, optic flow parameters, speed of flight, and other system level parameters; we did not explore the minimum detectable obstacle size in this work. For a discussion of why ER is more robust than optical flow for obstacle avoidance and small object detection from a moving robotic platform, see Ref. 13.

The steering field is generated using concepts from the biologically inspired potential fields from Ref. 34. Specifically, the heading rate is obtained from

$$\ddot{\psi} = -b\dot{\psi} - k_g(\psi - \psi_g)(e^{-c_1 d_g} + c_2) + k_o \sum_o \mathrm{sgn}(\psi - \psi_o)(e^{-c_3|\psi-\psi_o|})(e^{-c_1(\mathrm{ER}_{\max}-\mathrm{ER})}) \tag{2}$$

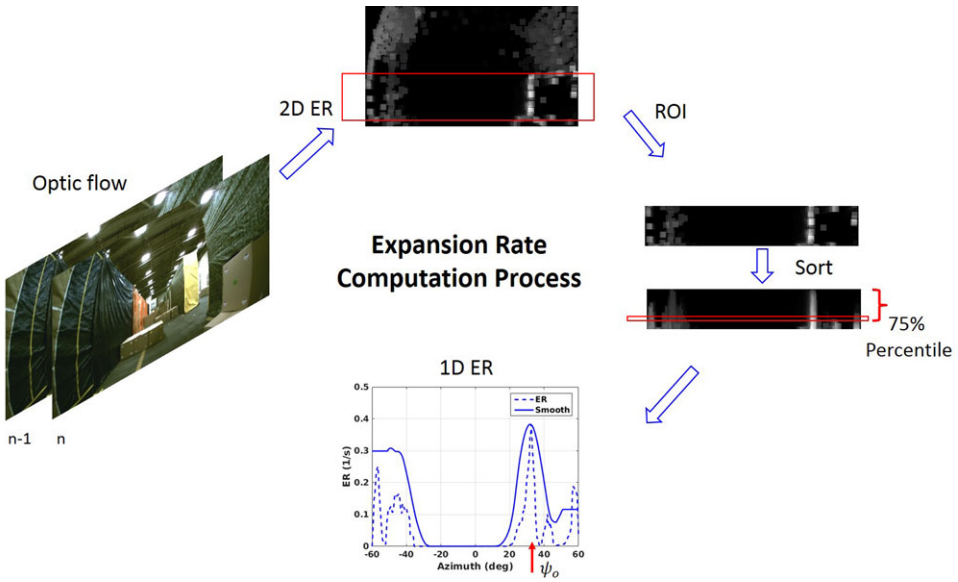

FIGURE 5 Expansion rate generation from motion vectors

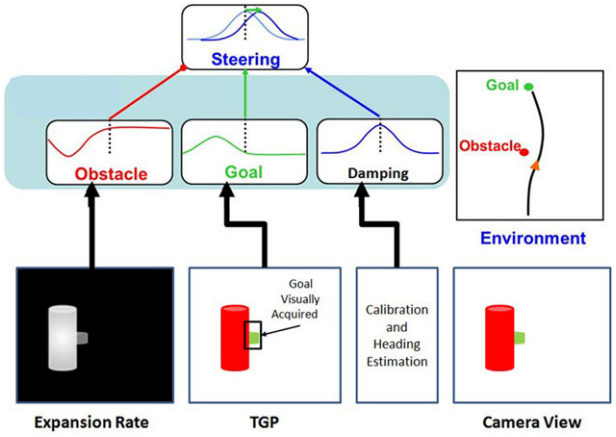

FIGURE 6 Steering field representation

where ψ is the heading direction (FOE), ψ_g is the goal direction, ψ_o is the obstacle direction estimated from ER, and d_g is the range to the goal. The constants b, k_g, k_o, c_1, c_2, and c_3 correspond to damping term, goal gain, obstacle gain, and tuning gains, respectively. For a detailed analysis of the motivation and role of each of these parameters, see Ref. 34.

Since the 1D ER representation is a direct measure of risk, it can be easily translated into a risk-based speed controller. If the mean value of the expansion rate overcomes a high-risk threshold, then the vehicle is commanded to slow down; if the mean ER is lower than a predefined low-risk threshold, then the vehicle is commanded to speed up, otherwise the vehicle will maintain its current speed; this is shown in Figure 7.

Note that the steering field controller is not designed to navigate out of all possible environments, rather it is designed to generate an instantaneous steering decision that minimizes collision risk. A higher level autonomy, which can react more slowly, is required to generate navigation plans that detect and avoid local minima via some other method such as arbitrary intermediate-goal waypoints.

3.3 | Goal detection

The implemented goal detection and recognition system is designed to operate without *a priori* training, and it uses a small number of templates to define the goal for a specific mission. In flight tests described in this article, a single image of the goal (a red or blue barrel) was provided *a priori*. Standard OpenCV normalized cross-correlation (or template matching)[35] functions were incorporated into an image search at multiple scales to provide a likelihood map for the goal position. If the likelihood in a given image region is greater than a detection threshold, then the goal object is considered to have been detected at that location and scale, with some confidence. Image location is converted to bearing, and image scale is converted to a rough range estimate. This goal range-bearing estimate is used by the executive planner to update the relative position filter and to subsequently update the goal field that is provided to the steering field controller.

The goal detection, recognition, and search models use luminance-invariant chromaticity channel features. After an extensive evaluation using data collected from different environments, the YUV color space was selected since it was demonstrated to be more robust to this approach of detection than RGB or HSV. A sample goal detection and template (blue barrel) can be seen in Figure 8.

4 | RESULTS

This section presents results of R-ADVANCE navigating in different indoor and outdoor environments while avoiding obstacles. R-ADVANCE has been tested in a warehouse interior with obstacles; outdoor environments with man-made obstacles such as buildings; outdoor environments with natural obstacles, i.e., trees; and outdoor

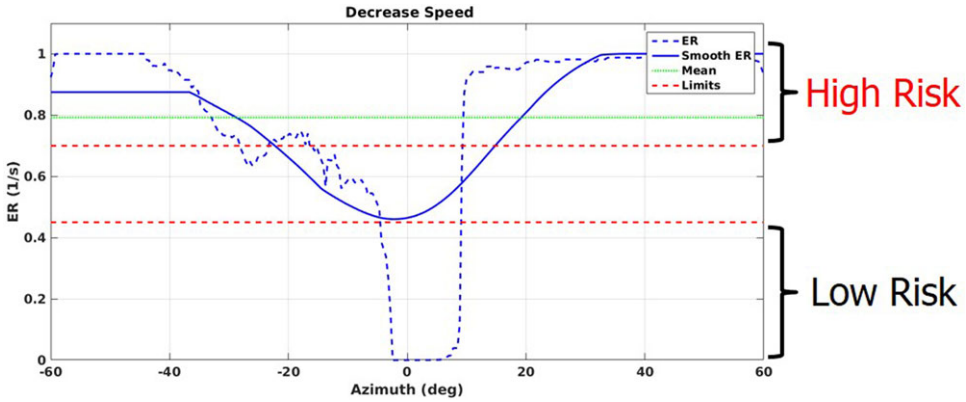

FIGURE 7 Risk-based speed control design

FIGURE 8 Goal detection using template matching: template of goal (left), detection of goal in environment (right)

open-space environments. Table 1 shows the different environments tested as well as the maximum speeds achieved during flight. For these missions, only the initial and final positions are known in advance, as shown in the overhead map in Figure 9. The location and sizes of intermediate obstacles and surroundings are generally unknown. For the purposes of this article, a successful mission is defined as the vehicle autonomously flying from takeoff position to within 10 m of the goal location without collision. Under some circumstances we flew missions where the locations of large fixed obstacles (such as buildings) were known in advance and incorporated into a multi-waypoint mission plan—this can significantly extend the capability of the system where reactive navigation alone is insufficient to find the goal object.

Video results of the system can be found at http://www.youtube.com/playlist?list=PLS01G8CWzRSv-G26YfMx8JkxgafHHqS--.

Key events of the missions will be highlighted next as they show how the individual components of the system work. As mentioned before, the vehicle is equipped with a collision-risk-based speed controller that makes it slow down as clutter increases. An example of this situation can be seen in Figure 10. The center section shows the images viewed by the vehicle and includes the ROI used for ER estimation. The ER image represents the environmental clutter including branches and trees, and therefore in the bottom section it can be seen that the 1D ER surpasses a threshold causing a slow down or decrease of speed, which can be seen in the top right plot.

Figure 11 shows a snapshot from a collision avoidance run that took place in a forest scenario in which the R-ADVANCE system simultaneously controls turn rate and regulates speed to prevent collision. It can be seen that the speed was commanded to 3 m/s and a negative turn rate to avoid the immediate tree shown to the right of the vehicle. This image is part of a longer trajectory that correctly navigated several tree obstacles in its path. The expansion rate that is superimposed on the images shows the two trees highlighted in white, which indicates a higher (immediate) collision risk than the rest of the environment.

The ER-based obstacle avoidance works regardless of whether obstacles are natural (trees) or man-made (buildings). Building obstacle avoidance can be seen in Figure 12. The expansion rate is

TABLE 1 Total distance from start and speeds achieved

Img #	Environment	Distance to goal (m)	Max speed (m/s)	Successful flights
1	Indoor warehouse	65	6	9
2	Outdoor avoiding building	172.8	10	4
3	Outdoor avoiding building and trees	243	7	1
4	Straight line to a building	357	8	3
5	Straight and 90° turn into building	128	12–15	5
6	Tree corridor	116	8	7
7	Long tree corridor	268	6–8	3
8	Straight into warehouse	189	8	1
8	Out the door of a warehouse	189	19	9
4	Straight long runs	500	12	2

FIGURE 9 Missions accomplished from top left to bottom right: 1-Indoor warehouse; 2-Outdoor avoiding building; 3-Outdoor avoiding building and trees; 4-Long straight line to building; 5- 90° turn to building; 6-Tree corridor; 7-Long tree corridor; 8-In/Out of warehouse

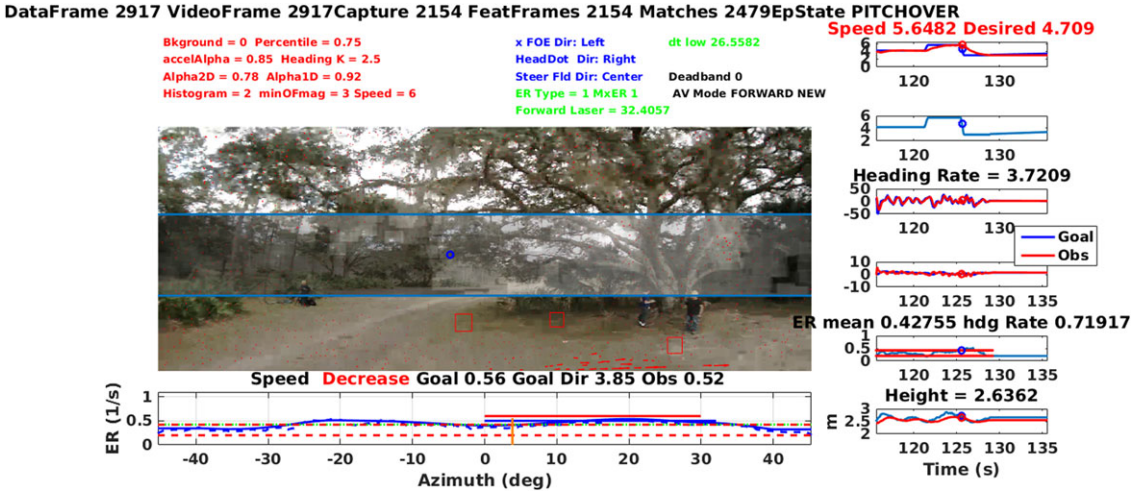

FIGURE 10 Speed control driven by the environment clutter. An image frame captured from the UAV Flea3 camera is shown with an overlay showing the ROI processed for ER. High ER is represented by white areas in the overlayed image. Below the image is the 1D ER representation computed from the image. Dotted horizontal lines show the slow-down (top) and speed-up (bottom) thresholds, and a vertical line shows the goal bearing. As can be seen, the clutter in this environment pushes the 1D ER representation above the slow-down threshold. The plots on the right, from top to bottom, are as follows: Speed desired (blue) vs. measured (red); Speed close up; Commanded heading rate; Components of steering field; ER over time with thresholds; and Height above ground

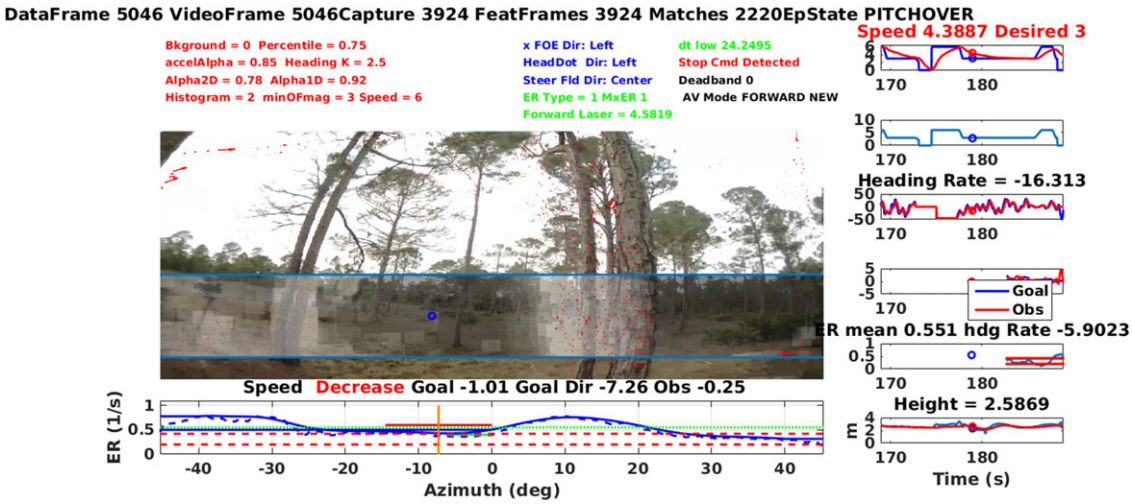

FIGURE 11 Obstacle avoidance in a tree forest highlighting the expansion rate on the closest trees

high on the right where it corresponds to the building and a negative heading rate indicates a left turn was commanded to avoid it.

A natural occurrence in the ER implementation is that openings become attractors as they do not contain any features and therefore the vehicle will tend to fly toward it, assuming the goal is in that direction. This behavior was demonstrated when a mission consisted of reaching a goal-object inside a building, as seen in Figure 13.

An example of the vehicle traversing through a forest at 10 m/s is shown in Figure 14, and a vehicle exiting a warehouse at 16 m/s is shown in Figure 15.

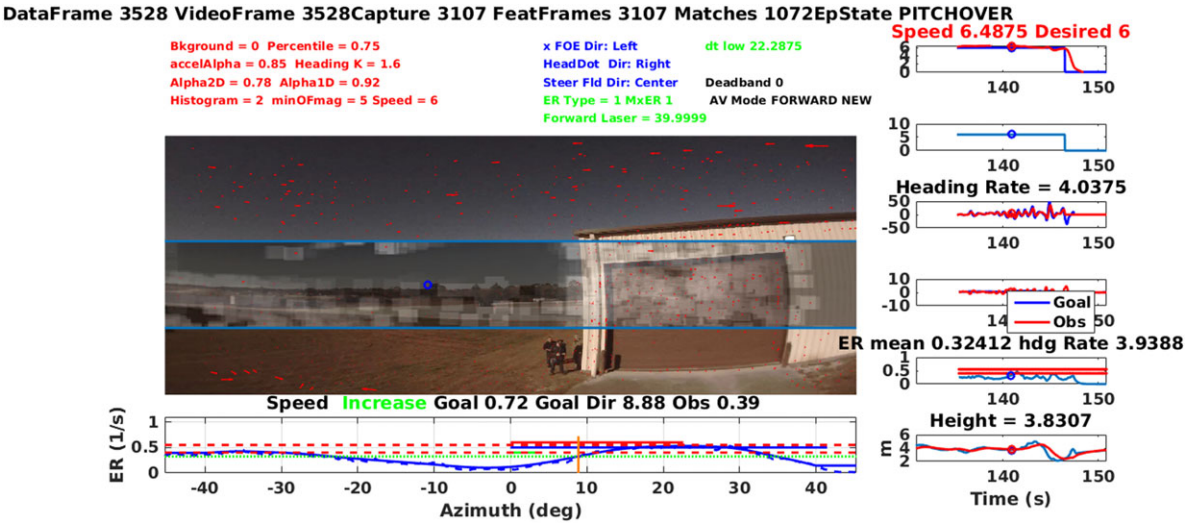

FIGURE 12 Vehicle avoiding the corner of the building when the goal is behind it, 7 m/s

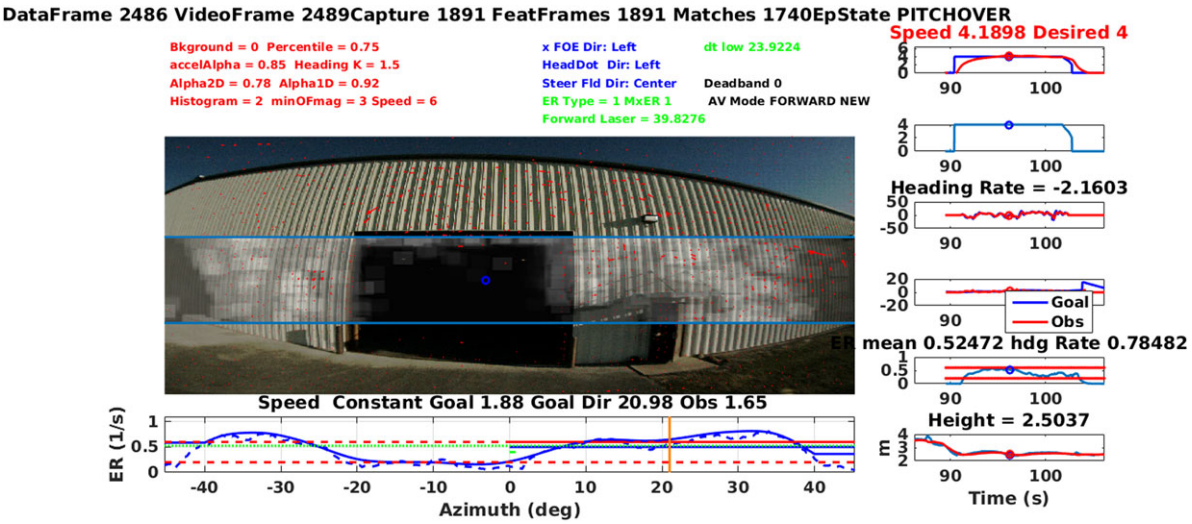

FIGURE 13 Warehouse entry: open door behaves like attractor, 5 m/s

5 | CONCLUSIONS

The R-ADVANCE system concept (based on ViSTARS[13]) was implemented on a micro-UAV on low-SWP hardware, and it was demonstrated to be capable of flying indoors and outdoors toward a target without the aid of GPS at speeds of up to 19 m/s. R-ADVANCE automatically avoids obstacles and controls speed based on overall collision risk while flying as fast as possible toward a visually detected goal object. System performance was robust to many factors, including the change of illumination when traversing from indoor to outdoor environments, and the different nature of natural and man-made obstacles. We are continuing to work on the overall reliability and robustness of the system. Below is a high-level summary of achievements and conclusions:

- Successfully completed missions covering distances of up to 500 m.
- Achieved autonomous flight speeds of up to 19 m/s.
- Successfully detected and avoided natural and man-made obstacles in indoor and outdoor unknown environments using expansion rate.
- Successfully implemented a reactive speed controller based on risk probability derived from ER.
- Successfully detected a goal object, and flew toward it, using cross-correlation search without the need of training data.
- For more complex and longer mission definitions, a more sophisticated executive planner will be needed.

6 | FUTURE WORK

Future work on R-ADVANCE will primarily consist of making the system more robust. This will include incorporation of more advanced visual odometry than that provided by the PX4Flow, updates to the obstacle avoidance and reactive control parameters, and a more robust

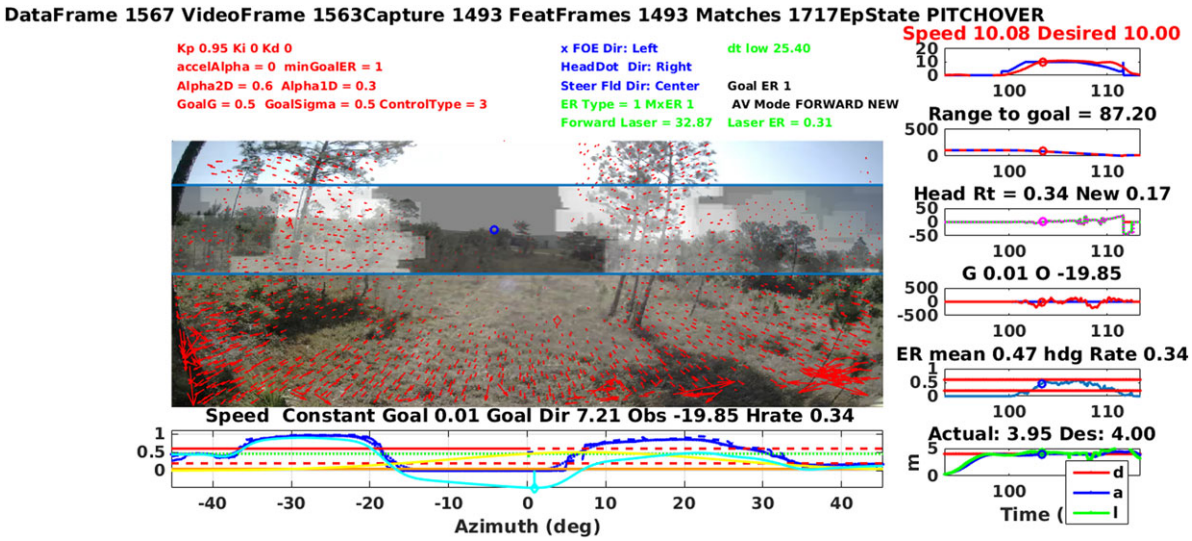

FIGURE 14 Vehicle flying through trees at 10 m/s

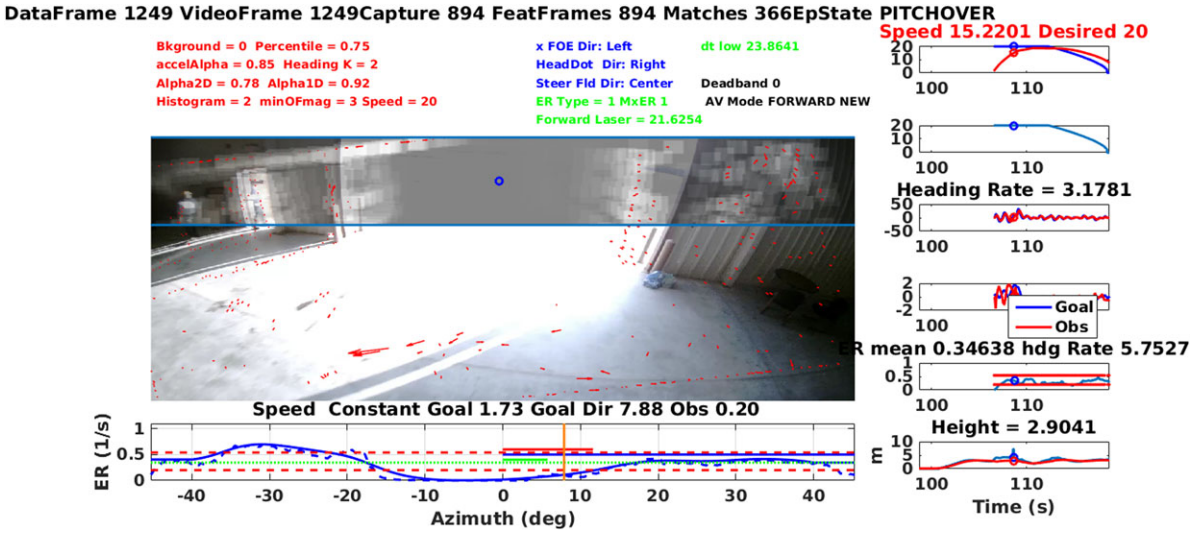

FIGURE 15 Vehicle exiting a warehouse at 16 m/s

high-level planner that incorporates visual verification of intermediate waypoints. Expansion rate, coupled with the steering field, was shown to be highly effective for reactive obstacle avoidance, but there were still cases when the ER magnitude was not strong enough to completely avoid a collision, and a more systematic way of tuning and balancing gains should be developed to avoid situations in which the goal and obstacle fields fight each other, creating oscillations and potentially resulting in collision.

ACKNOWLEDGMENTS

This work is supported by the Defense Advanced Research Projects Agency under Contract No. HR0011-15-C-0111. The views, opinions, and/or findings expressed are those of the author(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

ORCID

Hector D. Escobar-Alvarez http://orcid.org/0000-0001-8457-6071

REFERENCES

1. DJI Guidance (2017). Retrieved January 25, 2017, from http://store.dji.com/product/guidance.
2. Parrot SLAM Dunk (2017). Retrieved January 25, 2017, from https://www.parrot.com/us/business-solutions/parrot-slamdunk#parrot-slamdunk.
3. Intel RealSense R200 (2017). Retrieved January 25, 2017, from https://software.intel.com/en-us/RealSense/R200Camera.
4. Qualcomm Snapdragon (2017). Retrieved January 25, 2017, from https://developer.qualcomm.com/hardware/snapdragon-flight.
5. Ross S, Melik-Barkhudarov N, Shankar KS, et al. Learning monocular reactive uav control in cluttered natural environments. In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2013:1765–1772.

6. Richter C, Roy N. Towards high-speed autonomous navigation of unknown environments. In: *SPIE Defense+ Security*. International Society for Optics and Photonics; 2015:94671P–94671P.
7. Oleynikova H, Honegger D, Pollefeys M. Reactive avoidance using embedded stereo vision for mav flight. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2015:50–56.
8. Shen S, Mulgaonkar Y, Michael N, et al. Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In: *Robotics: Science and Systems*, vol. 1. Citeseer; 2013.
9. Bry A, Bachrach A, Roy N. State estimation for aggressive flight in gps-denied environments using onboard sensing. In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2012:1–8.
10. Richter C, Bry A, Roy N. Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In: *Robotics Research*. Springer; 2016:649–666.
11. Barry AJ, Tedrake R. Pushbroom stereo for high-speed navigation in cluttered environments. In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2015:3046–3052.
12. Browning NA. A neural circuit for robust time-to-contact estimation based on primate mst. *Neural Comput*. 2012;24(11):2946–2963.
13. Browning NA, Raudies F. Visual navigation in a cluttered world. In: Cristbal G, Perrinet L, Keil MS, eds. *Biologically Inspired Computer Vision: Fundamentals and Applications*. Weinheim, Germany: Wiley-VCH Verlag GmbH and Co. KGaA; 2015:425–446.
14. Byrne J, Taylor CJ. Expansion segmentation for visual collision detection and estimation. In: *2009 IEEE International Conference on Robotics and Automation*. IEEE; 2009:875–882.
15. Browning NA, Grossberg S, Mingolla E. Cortical dynamics of navigation and steering in natural scenes: Motion-based object segmentation, heading, and obstacle avoidance. *Neural Netw*. 2009;22(10):1383–1398.
16. Elder DM, Grossberg S, Mingolla E. A neural model of visually guided steering, obstacle avoidance, and route selection. *J Exp Psychol*. 2009;35(5):1501.
17. Zufferey J-C, Klaptocz A, Beyeler A, et al. A 10-gram vision-based flying robot. *Adv Robot*. 2007;21(14):1671–1684.
18. De Wagter C, Tijmons S, Remes BD, et al. Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2014:4982–4987.
19. Matthies L, Brockers R, Kuwata Y, et al. Stereo vision-based obstacle avoidance for micro air vehicles using disparity space. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE; 2014:3242–3249.
20. NVIDIA Jetson TK1 (2017). Retrieved January 18, 2017, from http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html.
21. Flea3 Camera (2017). Retrieved January 18, 2017, from https://www.ptgrey.com/flea3-usb3-vision-cameras.
22. Vector Nav 100 (2017). Retrieved January 18, 2017, from http://www.vectornav.com/products/vn100-rugged.
23. UM7-LT (2017). Retrieved January 25, 2017, from http://www.chrobotics.com/shop/um7-lt-orientation-sensor.
24. Lidar-lite V2 (2017). Retrieved January 25, 2017, from https://www.sparkfun.com/products/retired/13680.
25. PixHawk (2017). Retrieved January 25, 2017, from https://pixhawk.org/modules/pixhawk.
26. Shim I, Lee J-Y, Kweon IS. Auto-adjusting camera exposure for outdoor robotics using gradient information. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE; 2014:1011–1017.
27. Karpenko A, Jacobs D, Baek J, et al. Digital video stabilization and rolling shutter correction using gyroscopes. *CSTR*. 2011;1–2.
28. Harris C, Stephens M. A combined corner and edge detector. In: *Alvey Vision Conference*, vol. 15. Citeseer; 1988:10–5244.
29. NVIDIA VisionWorks (2017). Retrieved May 16, 2017, from https://developer.nvidia.com/embedded/visionworks.
30. Camus T, Coombs D, Herman M, et al. Real-time single-workstation obstacle avoidance using only wide-field flow divergence. In: *Proceedings of the 13th International Conference on Pattern Recognition*, vol. 3. IEEE; 1996:323–330.
31. Nelson RC, Aloimonos J. Obstacle avoidance using flow field divergence. *IEEE Trans Pattern Anal Machine Intell*. 1989;11(10):1102–1106.
32. Humbert JS, Hyslop AM. Bioinspired visuomotor convergence. *IEEE Trans Robotics*. 2010;26(1):121–130.
33. Franceschini N, Ruffier F, Serres J. Optic flow based autopilots: Speed control and obstacle avoidance. In: *Flying Insects and Robots*. Springer; 2009:29–50.
34. Fajen BR, Warren WH, Temizer S, et al. A dynamical model of visually-guided steering, obstacle avoidance, and route selection. *Int J Comput Vis*. 2003;54(1–3):13–34.
35. OpenCV NCC (2017). Retrieved January 25, 2017, from http://docs.opencv.org/2.4/doc/tutorials/imgproc/histograms/template_matching/template_matching.html.

How to cite this article: Escobar-Alvarez HD, Johnson N, Hebble T, et al. R-ADVANCE: Rapid Adaptive Prediction for Vision-based Autonomous Navigation, Control, and Evasion. *J Field Robotics*. 2018;35:91–100. https://doi.org/10.1002/rob.21744