J Intell Robot Syst (2010) 57:485–503
DOI 10.1007/s10846-009-9356-4

Development and Evaluation of a Chase View for UAV Operations in Cluttered Environments

James T. Hing · Keith W. Sevcik · Paul Y. Oh

Received: 1 February 2009 / Accepted: 1 August 2009 / Published online: 19 August 2009

Abstract Civilian applications for UAVs will bring these vehicles into low flying areas cluttered with obstacles such as building, trees, power lines, and more importantly civilians. The high accident rate of UAVs means that civilian use will come at a huge risk unless we design systems and protocols that can prevent UAV accidents, better train operators and augment pilot performance. This paper presents two methods for generating a chase view to the pilot for UAV operations in cluttered environments. The chase view gives the operator a virtual view from behind the UAV during flight. This is done by generating a virtual representation of the vehicle and surrounding environment while integrating it with the real-time onboard camera images. Method I presents a real-time mapping approach toward generating the surrounding environment and Method II uses a prior model of the operating environment. Experimental results are presented from tests where subjects flew in a H0 scale environment using a 6 DOF gantry system. Results showed that the chase view improved UAV operator performance over using the traditional onboard camera view.

Keywords UAV safety · UAV accidents · UAV training

J. T. Hing · K. W. Sevcik · P. Y. Oh (✉)
Department of Mechanical Engineering and Mechanics, Drexel University,
3141 Chestnut Street, Philadelphia, PA, USA
e-mail: paul@coe.drexel.edu

J. T. Hing
e-mail: jth23@drexel.edu

K. W. Sevcik
e-mail: kws23@drexel.edu

1 Introduction

Safe and efficient remote operation of a UAV in a cluttered environment requires that the pilot have a good sense of the state of the vehicle and the surrounding environment that the vehicle is operating in. Even in autonomous systems, it is still very important for the operator to know the state of the vehicle and its surroundings, especially when dealing with system faults. This awareness of the state of the vehicle and it's surroundings is called situational awareness(SA). The accepted definition of SA comes from [1] where it is broken down into levels. Level 1 SA is the perception of the elements in the environment within a volume of time and space. Level 2 SA is the comprehension of their meaning and Level 3 SA is the projection of their status in the near future. Situational awareness is effected by many factors. Current remote and autonomous systems are limited in what information is relayed from the vehicle back to the UAV pilot/operator. The operator's physical separation from the vehicle eliminates all motion feedback where as manned aircraft pilots utilize this motion to help in vehicle control. A UAV pilot's field of view is restricted due to limitations of the lens and positioning of the onboard camera. The limited field of view makes it difficult for the pilot to know the location of the extremities of the vehicle, which the authors believe to be critical knowledge when operating in a cluttered environment. The onboard camera view also requires a constant mental re-mapping of the environment by the pilot due to changing camera angles, which can also lead to vertigo. A pilot's understanding of the state of the vehicle, surrounding environment and mission status is solely reliant on the overwhelming visual representation of this information, many times leading to mental exhaustion. These limitations combined with a high workload lead to a lower situational awareness thereby increasing the chance for a mishap or accident.

Civilian applications for unmanned aerial vehicles (UAVs) will introduce these vehicles into cluttered near earth environments [2]. These are low flying areas typically cluttered with obstacles such as buildings, trees and power lines. More importantly, these areas are also populated with civilians. With the current accident rate of UAVs being significantly higher than that of commercial airliners [3], the idea of UAVs being operated in this type of environment is alarming. However, the potential of these vehicles to greatly benefit civilians demands that we evaluate what is necessary to improve the safety and operations of these vehicles in these environments.

The current state of the art UAVs are designed and operated to successfully complete tasks that commonly take place in higher altitude areas with very few obstacles to navigate around [4]. Most UAVs during a majority of these mission are operated under some sort of autopilot control. However, operation in cluttered environments requires fast and accurate obstacle avoidance algorithms, fast object recognition, and quick adaptation to changing conditions. Limitations of current UAV systems, specifically the current limits of artificial intelligence, can be overcome in these environments by keeping the UAV under the full control of a human pilot. However, control of the vehicle by the pilot is not enough, especially for operations in cluttered environments, as supported by the limited SA of current UAV pilots listed earlier.

In this paper, we investigate an approach to improving SA that utilizes sensor packages common on most UAV systems. The approach uses an onboard camera and

an inertial measurement unit to generate a mixed-reality chase view to the operator as seen in Fig. 1. There are two methods that we are developing to generate the mixed-reality chase viewpoint. The mixed-reality notion comes from the fact that the surrounding environment displayed to the pilot (outside of the onboard camera field of view) is a virtual representation. In method I, the surrounding environment is created by real-time mapping of features extracted from the onboard camera view. In method II, the surround environment is created using a prior model of the environment being operated in. A prior model could be constructed using geospatial digital terrain elevation data (DTED), satellite imagery, or prior manned or unmanned forward observer reconnaissance missions. For the chase view, the onboard camera images are still relayed to the pilot but are rotated, keeping the horizon level, and keeping the perspective consistent with the displayed chase viewpoint. This view allows the pilot to see the entire aerial vehicle pose and surrounding environment as if they were following a fixed distance behind the vehicle. The benefits of this viewpoint are an increased awareness of the extremities of the vehicle, better understanding of its global position in the environment, mapping of the environment, and a stable horizon (which helps to reduce the chance for vertigo as well).

The main contribution of this paper are the results from studying the differences when piloting a UAV in an cluttered environment while using a chase viewpoint versus using an onboard camera viewpoint. The results are obtained from experiments conducted at H0 (1:87) scale using a 6 degree of freedom (DOF) gantry that emulates an aircraft's translations and rotations. Also presented is the continuing work developing this approach for real world testing. The rest of this paper is

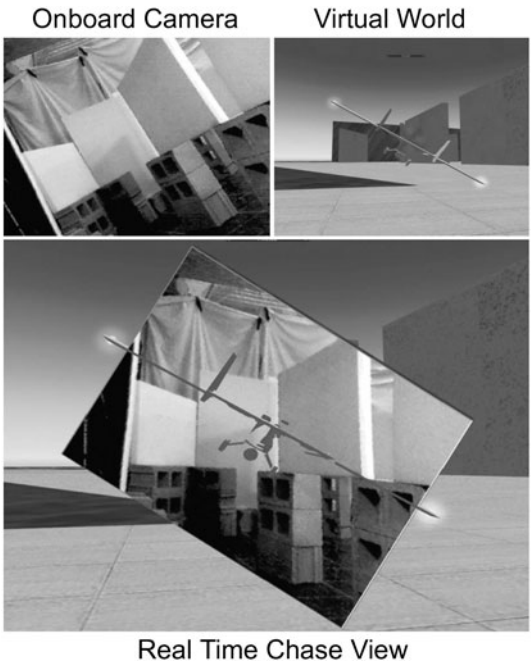

Fig. 1 The chase viewpoint during UAV operation in a cluttered environment consists of the real-time onboard camera image integrated into a virtual representation of the surrounding environment and aircraft pose

organized as follows: Section 2 gives some background on the previous work conducted in the area of improving situation awareness for UAV pilots. Section 3 presents the methods behind generating a chase view for UAV operation. Section 4 presents the experimental setup for evaluating UAV pilot skills in cluttered environments with different viewpoints. Section 5 presents the results from the study and Section 6 concludes the paper with a discussion and future work.

2 Previous Work

SA for operators of robotic ground and aerial vehicles has been investigated by a few researchers such as in [5]. In [5] it was reported that robots being operated at a post World Trade Center site were being operated with some sort of operator error 18.9% of the time due to poor interfaces and lack of functional presence. Research such as this and others have lead to proposals on ways to improve UAV pilot situational awareness. In prior work, the authors of this paper investigated the use of motion platform technology to relay motion cues to a UAV pilot [6]. While the method showed potential, the current motion platform technology is very large and expensive for a majority of the potential civilian UAV operators. Other researchers have investigated new designs for heads up displays [7], adding tactile and haptic feedback to the control stick [8, 9] and larger video displays [10]. Synthetic vision, in recent years, has been studied and shown to improve situational awareness for remotely piloted vehicles [11]. Synthetic vision displays to the operator a far distance exocentric view of the UAV with a virtual representation of the terrain based on a database of elevation maps. The method requires prior knowledge of the terrain/elevation and does not include obstacles other than the natural terrain data. Synthetic vision is mostly used to depict the planned trajectory from a 3D perspective for support in guidance and control. Also related to what we are proposing is the work conducted by [12, 13] where real world current data is used for generating virtual views.

Sugimoto et al. [12] and Nielsen et. al. [13] developed methods for viewing remotely operated ground vehicles from outside the vehicle; “Time Follower’s Vision” for [12] and tethered position in [13]. Both methods produced a viewpoint that allowed an entire visualization of the vehicle pose and the environment directly surrounding the vehicle itself. Both works presented studies showing that their methods improved remote operation of the vehicle in both speed of operation and accuracy of vehicle positioning. In [12] however, they did not use any sort of mapping of the environment, leaving this up to the memory of the operator. In [13] they generated a 2-D map of the environment as the vehicle drove around, using a laser range finder and SLAM. This map was relayed to the operator in the tethered view showing walls as slightly raised obstacles.

A major challenge faced when adapting concepts created for ground vehicles, as in [12, 13], to air vehicles is that air vehicles are capable of much more movement than ground vehicles. In cluttered and urban environments, UAVs can undergo large three dimensional translations and rotations. Another challenge is that, obstacles can not be represented by infinitely high walls (often used in 2D ground vehicle maps) as UAVs can fly around, above, and in the case of overpasses, below obstacles. UAVs, especially those flown in urban environments, will be small so they can maneuver

between obstacles with relative ease. The small size limits the payload capacity of the vehicle. Laser range sensors, like those used in [13], can be too heavy to add to a typical UAV sensor suite that already includes an IMU, GPS and an onboard camera.

Closely related to the work presented by the authors of this paper is the work conducted by [14, 15]. In [14], during high altitude search tests, they used real-time video data from the UAV and augmented it with pre-loaded map data (satellite imagery). The onboard camera view is rotated to match the preloaded terrain map and a silhouette of the UAV is displayed on the map showing its heading. The preloaded terrain map does not change it's orientation. This would have some negative effects in piloting the vehicle in terms of remapping of the controls, for example when the UAV is flying south on the map, pulling left on the joystick makes the UAV go to the right. However, the purpose of the research was not to improve operator control (the UAV flew at a high altitude autonomously through waypoints), but to improve the situational awareness of the observer conducting the search task. Their results showed that the augmented image helped the observers comprehension of the 3D spatial relationship between the UAV and points on the earth. In [15] they investigated the effects of displaying a simplified "wing-view" of the UAV to the operator that shows roll and altitude of the aircraft. The motivation was that UAV operators are standing on the ground and a display with a moving horizon may confuse users not trained as pilots. This display helped with the operators in the understanding of the instantaneous relationship between the UAV and the world.

The authors of this paper believe that the concepts investigated in [12–15] could be adapted and developed such that they can be applied to UAV operation in cluttered environments. Utilizing the IMU and onboard camera, the authors of this paper show two methods for generating a chase view point for UAV pilots.

3 Methods Towards Generating Chase Viewpoint

UAVs operating in urban and cluttered environments will most likely be limited to smaller back-packable and hand launchable vehicles that enable quick maneuvering and access to small spaces. With limited payload, choosing an optimal sensor suite is extremely important. The ultimate goal is to gather all data about the state of the vehicle and information from the surrounding environment using as few sensors as possible.

There are two methods presented in this paper for generating a chase viewpoint. Method I utilizes an onboard camera and GPS to generate a 3D map of the environment. Method II utilizes the onboard GPS of the aircraft and prior knowledge of the operating environment to generate a surrounding 3D map. The advantage of Method I is that a map is created based on a very recent interaction with the environment and can be used without prior knowledge of the operating area. It can also be adapted to work in areas without GPS availability by finding vehicle state information from structure from motion methods. Method I however comes at a cost of computation power, which limits the speed at which the UAV is allowed to fly in the environment. Method II allows for much faster flight as the environment is already mapped. Should the environment change, the pilot will be forced to mentally remap the surrounding environment during the flight using the onboard camera view.

3.1 Method I

A chase viewpoint requires three dimensional measurements of the surrounding environment and accurate knowledge of the state of the vehicle. Researchers are currently working on methods to gather this information from only one onboard camera [16, 17] using Structure from Motion (SFM) methods. The added benefits of this is that UAVs can be smaller, and the vehicle is capable of map building in areas with no GPS signal. As these methods are currently computationally expensive, the authors of this paper chose to use information from an onboard IMU, GPS, and camera for the initial work toward developing the chase viewpoint. The technique for Method I is presented in the following sub sections.

3.1.1 Feature Detection and Tracking

Creating a map of the surrounding environment from the onboard camera view requires that three-dimensional information be extracted from multiple two-dimensional camera images. Features in each image must be found and tracked from frame to frame. Based on recommendations from [18], the authors use a 7×7 feature detection window and calculate the spatial gradient matrix, H, as the window scrolls through the camera image.

$$H = \Sigma \begin{vmatrix} (\delta I/\delta x)^2 & (\delta I/\delta x)(\delta I/\delta y) \\ (\delta I/\delta x)(\delta I/\delta y) & (\delta I/\delta x)^2 \end{vmatrix} \tag{1}$$

where I(x,y) is the gray level intensity and the summation is through the feature window. If the eigenvalues of H are greater than a chosen threshold then that particular area of the image is chosen as a feature point to track. Features are chosen such that they are the strongest features in the image, don't overlap, and only a set number of features desired by the user are kept.

Tracking of the feature points is conducted using a pyramidal implementation of the Lucas Kanade feature tracker (KLT)[19]. The pyramidal implementation allows for much larger movement between two images. Currently the authors are using a 3 level pyramid which can track pixel movement 8 times larger than the standard Lucas Kanade tracker. In a traditional pyramidal KLT, feature points are chosen in the highest level of the pyramid. This however did not produce desired results. As such, the authors proceeded with the following: First, features are detected on the highest resolution image which is currently at 640×480 (onboard camera resolution). A 5×5 gaussian blur is used before each re-sampling of the image all the way to the third level which is 80×60 resolution. The centroids of the chosen features are mapped to the location on the third level. For frame J to K, the previous and current onboard camera image respectively, the following calculations take place over 10 iterations with iteration i starting at 1:
First an image difference $\delta I(x, y)$ is calculated:

$$\delta I_i(x, y) = J^L(x, y) - K^L\left(x + g_x^L + v_x^{i-1}, y + g_y^L + v_y^{i-1}\right) \tag{2}$$

where for level 3 (L = 3), the initial guess g_x, g_y is zero and the iteration guess $v^0 = (0, 0)$. Then the image mismatch vector b_i is calculated for the feature window:

$$b_i = \Sigma \begin{vmatrix} \delta I_i(x, y) I_x(x, y) \\ \delta I_i(x, y) I_y(x, y) \end{vmatrix} \tag{3}$$

The optic flow η^i is then calculated:

$$\eta^i = H^{-1} b_i \tag{4}$$

And the guess for the next iteration becomes:

$$v^{i+1} = v^{i-1} + \eta^i \tag{5}$$

After the iterations are complete the final optic flow d^L for the level is:

$$d^L = v^{10} \tag{6}$$

The guess for the next lower pyramidal level g_x, g_y becomes:

$$g_x, g_y = 2\left(g^{L-1} + d^L\right) \tag{7}$$

And the process repeats until the final level (L^0), the original image, is reached. The final optic flow vector d is:

$$d = g^0 + d^0 \tag{8}$$

And the location of the tracked feature on image K is:

$$K(x, y) = J(x, y) + d \tag{9}$$

The tracking (50 features) is at sub pixel resolution and is currently running at 10 FPS on a 2.33 GHz dual core machine.

3.1.2 Reconstruction and Mapping

For the initial development, we are utilizing a simulated environment that we modeled in the flight simulation package X-Plane from Laminar Research as seen in Fig. 2. Since the authors chose to use an IMU and GPS along with the camera,

Fig. 2 Environment (Drexel University Campus) created in X-Plane for testing feature tracking and reconstruction. Initial textures were of grid patterns for easier development during the initial stages

structure from motion methods are not needed and the 3-Dimensional locations of the feature points can be found through triangulation. The extrinsic parameters for the camera are extracted from GPS and IMU measurements in the X-Plane simulation. The intrinsic parameters of the camera are calculated prior to the tests using multiple images of a known grid pattern. Calibration tests found the focal length for the camera in the X-Plane environment to be 320.469 mm. Each feature point is stored in its initial frame and then tracked. If the feature point is successfully tracked for 5 frames, as seen in Fig. 3, it is used in the reconstruction algorithm. The 5 frame difference was chosen to allow a greater distance between the two camera images before reconstruction is run. The global frame of reference is chosen such that the axes lie on the latitude (Y), longitude (X) and altitude directions (Z) of the simulated environment. The origin of the axes are located in the simulated world where the vehicle is initially spawned. The distance to the aircraft camera from the global reference frame is calculated from GPS and IMU values. Locations of feature

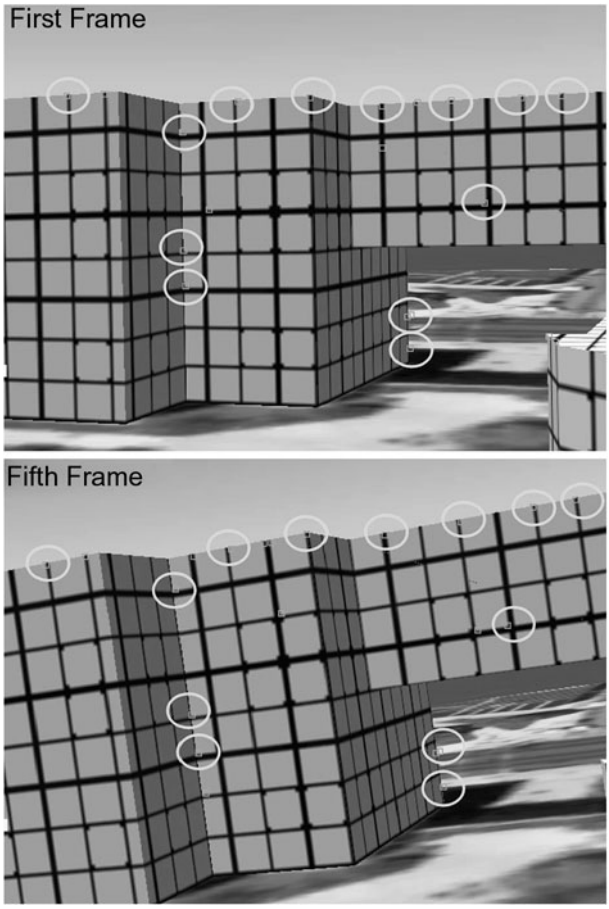

Fig. 3 Feature tracking across multiple frames. Features detected are surrounded by a *small box*. The tracked features used in reconstruction are highlighted by *circles*. The frames contain a rotated view (aircraft is rolling) side of a building at Drexel. The texture of the walls were created with a grid pattern to ease the use of feature detection/tracking for initial development

points in the camera image plane are transformed to the global reference frame using the following rotation and translation matrices:

$$\begin{aligned} R_{1,1} &= cos(\alpha)cos(\gamma) - sin(\alpha)sin(\beta)sin(\gamma) \\ R_{2,1} &= sin(\alpha)cos(\gamma) + cos(\alpha)sin(\beta)sin(\gamma) \\ R_{3,1} &= -cos(\beta)sin(\gamma) \end{aligned}$$

$$\begin{aligned} R_{1,2} &= -sin(\alpha)cos(\beta) \\ R_{2,2} &= cos(\alpha)cos(\beta) \\ R_{3,2} &= sin(\beta) \end{aligned} \tag{10}$$

$$\begin{aligned} R_{1,3} &= cos(\alpha)sin(\gamma) + sin(\alpha)sin(\beta)cos(\gamma) \\ R_{2,3} &= sin(\alpha)sin(\gamma) - cos(\alpha)sin(\beta)cos(\gamma) \\ R_{3,3} &= cos(\beta)cos(\gamma) \end{aligned}$$

$$T = \begin{vmatrix} F_d cos(\beta)sin(\alpha) + Lon. - Lon.of\,Origin \\ F_d cos(\beta)cos(\alpha) + Lat. - Lat.of\,Origin \\ F_d sin(\beta) + Alt. - Alt.of\,Origin \end{vmatrix} \tag{11}$$

where α is the camera heading angle, β is the camera pitch angle, γ is the camera roll angle, and F_d is the camera focal length.

Reconstruction proceeds as follows:

Following Fig. 4, the line running through the camera frame, C, and the feature point, P, on the image plane to the feature point in the global frame is:

$$l = C_L + a(P_L - C_L) \tag{12}$$

$$r = C_R + b(P_R - C_R) \tag{13}$$

where a and b are values between 0 and 1 representing the length of vectors l and r respectively between C and P.

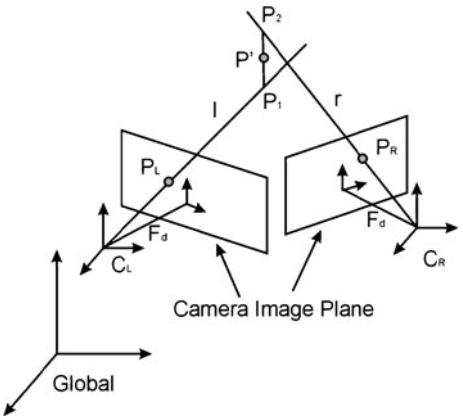

Fig. 4 Camera reconstruction geometry. Due to noise in the measurements, rays passing through the feature in the first and second camera image plane may not intersect. The midpoint of the closest point between the two rays is taken as the feature measurement

Ideally the two lines would intersect at the global location of the feature point, P, but due to noise in the measurements, they may not intersect. Therefore, it is determined that the feature point lies in the midpoint, P', between the line segment that is perpendicular to both of the rays.

$$P_1 = C_L + a_o(P_L - C_L) \tag{14}$$

$$P_2 = C_R + b_o(P_R - C_R) \tag{15}$$

$$P' = P_1 + 1/2(P_2 - P_1) \tag{16}$$

where ao and bo represent the values of a and b where the line P' crosses the l and r vectors respectively.

The orthogonal vector, w, to both lines, l and r, is:

$$w = (P_L - C_L) \times (P_R - C_R) \tag{17}$$

Therefore, the line going through P_1 to P_2 is:

$$P_2 = P_1 + c_o w \tag{18}$$

The unknowns a_o, b_o, c_o are found by solving the following equation:

$$a_o(P_L - C_L) - b_o(P_R - O_R) + c_o w = C_R - C_L \tag{19}$$

Currently the method is run without any filtering of the data so the results are somewhat noisy as seen in Fig. 5. The method up to this point runs at approximately 6 frames per second on a 2.33 GHz dual core Windows laptop. However, the programming code has not been optimized and the desired operation speed is to be no less that 10 frames per second. The following steps presented for Method I have not been completed as of yet due to current computation costs but are mentioned to build further understanding of the final goal.

Adapting a method similar to that presented by [20] we will create the three dimensional map of the environment from a single camera viewpoint. This map will then be used in the chase view perspective of the UAV pilot. What the authors of [20] do differently from a number of single camera map making algorithms is that they merge feature points into planar regions for use in SLAM. The benefits of this is that it dramatically reduces the number of stored feature points needed to create a map.

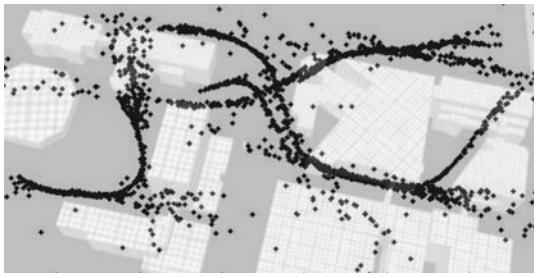

Fig. 5 Top down view of raw (non-filtered) reconstruction of feature points with flight environment overlayed over the data. Most data points far away from building edges are points reconstructed from features detected on the ground

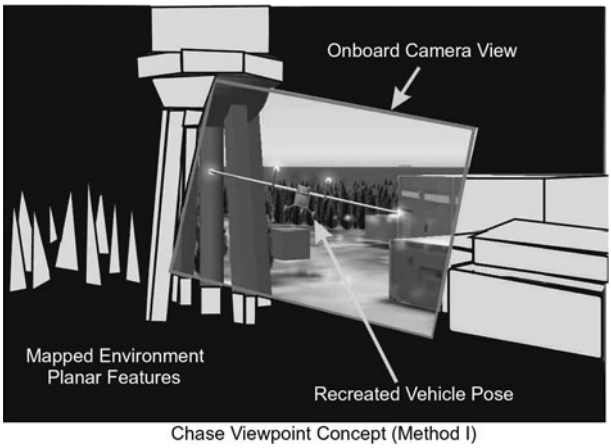

Fig. 6 Conceptual graphic showing the chase viewpoint during UAV operation in a cluttered environment

Much of urban terrain contains rectangular buildings. Therefore, many detected features can be turned into planar regions that represent building walls and rooftops. Once the mapping is completed, the chase viewpoint can be generated by integrating the UAV onboard camera view with the UAV perspective of the generated map. Also, a virtual view of the UAV will be displayed in the map, generated using IMU data and its relative size in the displayed map, along with the real time onboard camera footage which will be rotated (and warped if necessary) to match the chase view perspective. This concept can be seen in Fig. 6. This method of generating the chase view allows for a current map of the environment to be relayed to the operator at the expense of high computation requirements and limited flight speed.

3.2 Method II

As stated earlier, Method II requires much less computation during the flight as the operating environment is modeled prior to the flight. Again, one can easily generate such models from DTED data, satellite imagery and forward-observer reconnaissance. In a few applications, the environment will stay relatively static which makes Method II valid. For this paper, X-Plane flight simulation software is used to model the UAV operation environment during flight tests. Aircraft position in the modeled environment is updated by GPS from onboard the UAV. The onboard camera view is rotated based on the roll angle received from the onboard IMU and surrounded by the simulated environment.

4 Experiment Setup

To validate efforts toward generating a chase viewpoint for UAV pilots, experiments were setup to test pilot skills operating in a cluttered environment using the current onboard camera viewpoint and a generated chase view point. The ideal scenario is to have a chase-view of the actual environment built from the real sensor data. Method I is the work we have done toward that goal. However, results are noisy and the update rate is slow. To evaluate the utility of a chase view, we conducted

tests using an onboard camera fused with a pre-built 3D model. For the experiment results presented later in this paper, Method II is used.

4.1 Hardware

Field testing at the current stage of the project is risky and requires a long process of approvals to operate a UAV in restricted airspace. Tests using only a flight simulator would help validate the design notion but it is difficult to simulate the mechanical systems/sensors used in real world tests and environmental conditions. Such examples would be the camera visuals, actuator responses, lighting effects, fog and rain. Because of these reasons, the authors took advantage of the Systems Integrated Sensors Test Rig (SISTR) facility at Drexel University to conduct flight experiments in a scaled environment with actual UAV system hardware. SISTR was constructed with support from the U.S. National Science Foundation for the design and testing of UAVs and UAV sensor suites. SISTR is a 3 degree of freedom (DOF) gantry system with a workspace of $18' \times 14' \times 6'$ [21]. To match the size of a reasonable real world UAV test environment, SISTR's workspace represented an H0 scale (1:87) environment as seen in Fig. 7. The flight environment consisted of narrow corridors that can be representative of corridors between large buildings in an urban environment.

SISTR's end effector is used to represent the location of the aircraft inside of the scaled environment. Aircraft dynamics during the experiments are handled by a flight simulation package and the H0 scaled translational position of the aircraft is relayed from the flight simulator to SISTR's controller via User Datagram Protocol (UDP) at a rate of 20HZ.

The aircraft's angular positions are commanded by the experiment subject (pilot) via a joystick. The resulting angular position of the aircraft, generated by the flight simulator, is relayed to a 3 DOF yaw, pitch and roll (YPR) unit attached to SISTR's end effector as seen in Fig. 8. The YPR unit was specifically designed such that it represented the Euler angles of the aircraft; yaw is applied first, then pitch, then roll. It was also designed to have a small footprint due to operation in a scaled environment. A 640×480 resolution camera with 35 degree field of view, seen in Fig. 8, was attached to the YPR unit. The images from the camera represented the onboard camera view from the aircraft and were relayed to the experiment subject (pilot) at a rate of 30 frames per second during onboard camera tests and 10 FPS during chase view tests.

4.2 Software

Aircraft dynamics and the virtual environment are generated using a commercial flight simulator software known as X-Plane (also mentioned earlier in Method I). X-Plane incorporates very accurate aerodynamic models into the program based on blade element theory and allows for real time data to be sent into and out of the program [22]. X-Plane has been used in the UAV research community, mostly as a visualization and validation tool for autonomous flight controllers [23]. In [23] they give a very detailed explanation of the inner workings of X-Plane and detail the data exchange through UDP. During the experiment, flight commands are input into X-Plane by the subject via a joystick and X-Plane generates and sends the translational

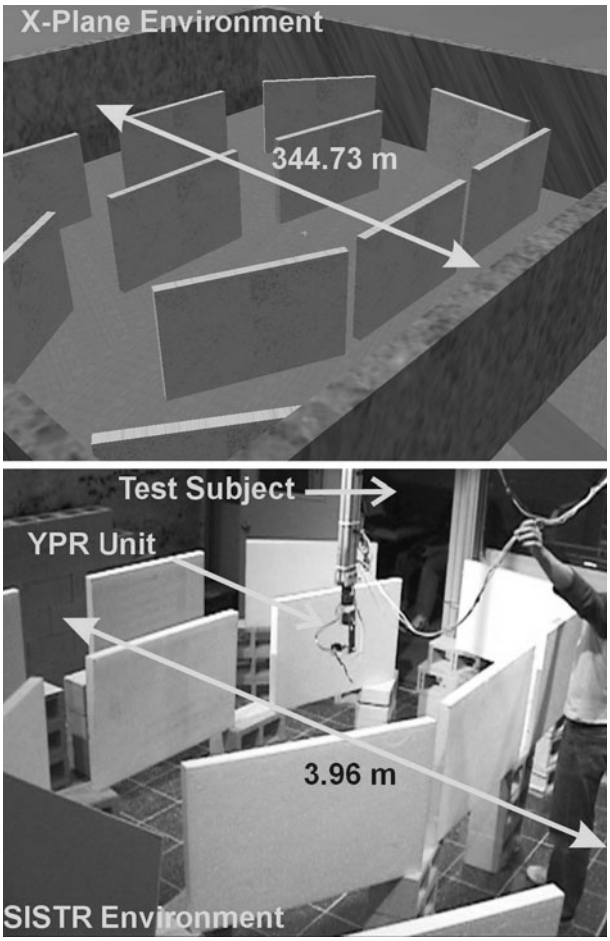

Fig. 7 Comparison showing the real world scale flight environment with the H0 scale (1:87) SISTR environment. The *boards* create narrow corridors representative of flight between large buildings in an urban environment

and angular positions through UDP to the SISTR controller. X-Plane is also used during the chase view experiments to render the surrounding virtual view of the aircraft environment based on a prior environment model. The H0 scale environment in SISTR was built to match the full scale corridor environment we created in X-Plane. The optics of the onboard camera are accounted for by adjusting the aspect ratio in X-Plane so that the virtual environment matches up with the onboard camera view.

A UAV model was created that represents a real world UAV, known as the Mako, currently in military operation. The Mako, as seen in Fig. 9, is a military drone developed by Navmar Applied Sciences Corporation. It is 130 lbs and has a wingspan of 12.8 ft. It is very similar in size and function to the more popularly known Raven and Pioneer UAVs. This UAV platform was ideal as NAVMAR is about 10 miles of Drexel, thus it could be validated, with quick feedback on the model and our notional concept, by veteran Mako pilots. For safety reasons, the simulated version of the Mako was modified so it was lighter weight with less horsepower effectively

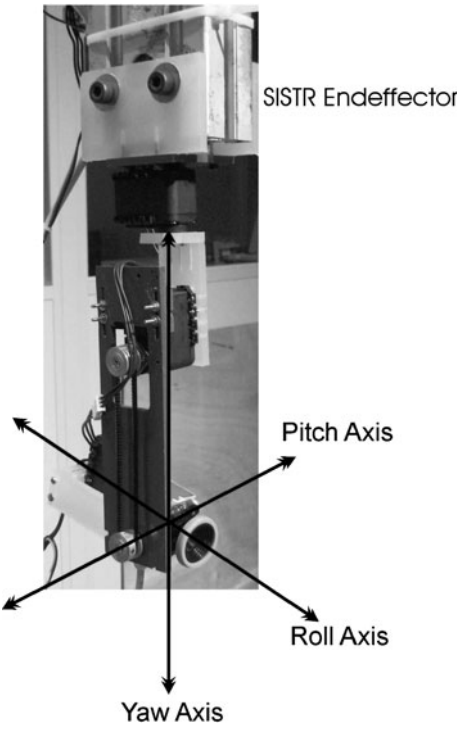

Fig. 8 Yaw, pitch and roll unit used to recreate the angular position of the aircraft inside of SISTR. The unit is designed based on the Euler angles of the aircraft. Yaw is applied first, then pitch, then roll

decreasing it's cruise speed to 30 miles per hour in the simulation which corresponds to 6 in./s in SISTR motion at H0 scale.

4.3 User Interface

The user interface was created using Visual C#. The program handled the visual presentation to the user and also the communication between X-Plane and SISTR. The program collected translational and angular position data from X-Plane, converted it to H0 scale and then transmitted it through UDP to SISTR at 20 Hz. During onboard camera tests, only the onboard camera view was shown to the pilots during flights through the environment as seen in Fig. 1(onboard camera). During the chase view tests, the program displayed to the pilot 3 items:

1. Rotated onboard camera view so the horizon stays level

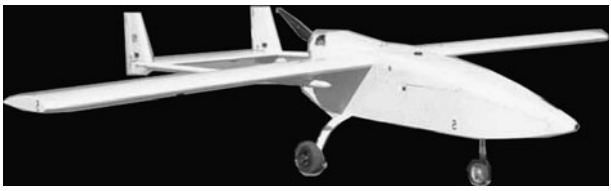

Fig. 9 The MAKO UAV developed by NAVMAR was modeled in X-Plane and used for experiment flights

2. Virtual view of the surrounding environment based on aircraft location and prior model of the environment
3. Virtual representation of the aircraft pose to scale with the onboard camera view and surrounding environment

These items, seen in Fig. 1 are relayed in real time to the pilot. The onboard camera image is rotated by applying a simple rotation about the center of the image based on the roll angle of the aircraft.

4.4 Procedure

Prior to the tests, subjects were given time to fly the Mako in an open environment in X-Plane under both simulated onboard camera view and chase view. This allowed them to become familiar with the controls and to get a feel for the response and size of the aircraft. When the subjects felt comfortable with the controls, the experiments began. As seen in Fig. 10, the subjects were placed in a room, separated from the experiment environment, with a 52″ monitor from which to view the user interface. Subjects underwent multiple tests where they flew the aircraft from an onboard camera view or a chase view. During onboard camera tests, the subjects were shown only the raw view from the camera and asked to fly through the corridors of the environment while keeping a safe distance from the walls and keeping the aircraft as stable as possible. During the chase view tests, the subjects were shown the chase view and asked to fly through the corridors with the same emphasis on safe distance and stability. During each test, aircraft translational and rotational positions were recorded. If the subject crashed into the corridor walls, they were asked to continue their flight through the corridors so data collection could continue. The walls of the SISTR environment were designed to easily collapse under contact. The walls in the X-Plane environment were designed to allow the plane to pass through. After each test, subjects were given a survey on their thoughts of the different modes during the flight tests.

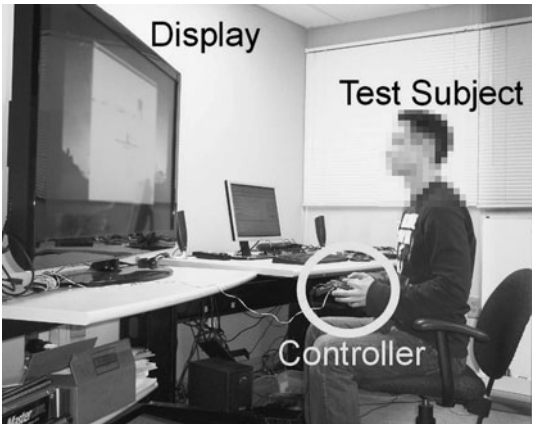

Fig. 10 View of the operator station for the experiments. The operator sits inside of a room separated from SISTR. The 52” television displays the different views for each test. Control of the aircraft (SISTR endeffector) is provided by inputs from the operator using the controller

5 Results and Discussion

Seven subjects were used for initial validation of the chase view concept. Each subject varied in flight simulator experience from no experience to 5 years worth of recreational use. Shown in Fig. 11 are the best flight paths, out of all the tests, achieved for each subject using a chase view and using an onboard camera view. While using the onboard camera view, subjects showed much more of an oscillatory movement than while using the chase view. This can be attributed to two issues. During the onboard camera view tests, due to the smaller field of view, subjects would continue to turn to bring the walls into view so to establish their position in the environment. The second issue was a slight lag in the response of the camera servos and gantry to the subject's commanded desired positions similar to lag commonly experienced in current UAV field operations. This caused some subjects to overcompensate in the controls which led to increased oscillations in the flight.

Since the goal was to keep the aircraft as stable as possible, angular accelerations were recorded to quantify how well the subjects were able to do this. Table 1 shows the average angular acceleration (low angular acceleration is representative of stable flight) of each subject during trials using the chase view and trials using the onboard camera view. Due to the lack of sample size, a statistical significance was not calculated but the trend in the data leads to the conclusion that the chase view decreases the angular accelerations commanded during the flight. This is significant as quick turns under normal UAV operations can induce high stresses on the vehicle leading to accelerated wear and tear, which in turn can lead to catastrophic failure. Interesting to note, some subjects such as subject 2 and 3, did not show a dramatic decrease in the angular accelerations when switching from onboard camera view to a chase view. This shows that the view did not help the subject decrease the amount of movement they were commanding to the vehicle. However, if you look at the flight paths, the chase view did accomplish a better path through the environment. In the case of subject 3, chase view was the only way that they were able to get through the majority of the course. Subject 2 and 3 both had little to no prior flight simulator

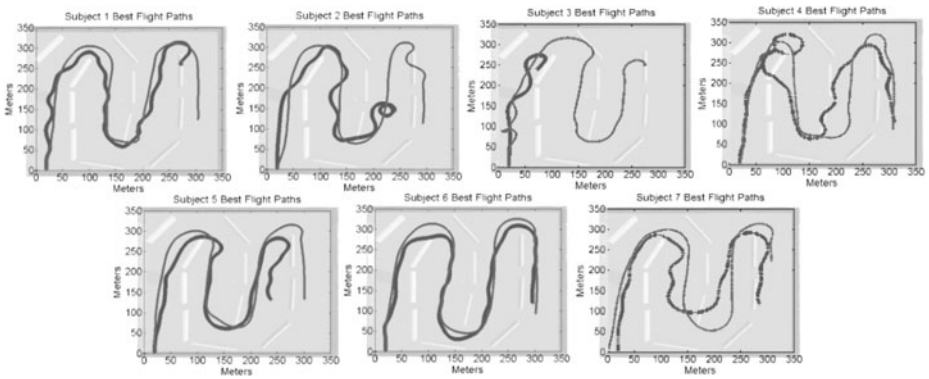

Fig. 11 The best flight path results for each of the 7 subjects using chase view and using the onboard camera view. The flight environment is overlayed on top of the graph. The *thin line* is the flight path using chase view, the *thick line* is the flight path when using the onboard camera view

Table 1 Average angular acceleration during chase view and onboard camera view tests

Subject	Yaw axis (deg/s^2)	Roll axis (deg/s^2)	Pitch axis (deg/s^2)	Magnitude (deg/s^2)
1 Onboard	30.00	84.56	29.65	94.52
1 Chase	22.88	36.72	20.38	47.97
2 Onboard	17.57	34.16	23.63	45.10
2 Chase	15.12	37.70	12.80	42.75
3 Onboard	29.08	56.03	36.67	73.17
3 Chase	32.06	45.87	28.95	63.36
4 Onboard	46.22	69.23	32.47	89.83
4 Chase	31.44	50.06	14.18	60.82
5 Onboard	22.95	48.95	18.86	57.29
5 Chase	7.53	23.02	8.41	25.64
6 Onboard	14.86	41.87	13.91	46.57
6 Chase	5.05	13.57	5.66	15.57
7 Onboard	28.26	36.39	11.99	47.88
7 Chase	4.69	13.61	6.47	15.78

experience before the tests (other than the warm up period). It is therefore believed that the lack of change in angular accelerations between views can be attributed to their limited understanding of how the controls effect the aircraft's flight.

The oscillatory motion is much easier to observe in Fig. 12 which shows the angular positions of the aircraft during an example of a subject's flight using the onboard camera view and chase view. During the onboard camera view tests, the subjects tended to move through a larger angular range and at a higher frequency than during the chase view tests. Most subjects after the tests stated that the chase view was much easier to operate with. For some subjects, the onboard camera view was so disorienting that they were unable to complete the course (even with crashing through barriers) after multiple tries. This was more common among subjects who had very little to no prior flight simulator experience. All of these subjects however were able to complete the course using the chase view within 2 trials. The example results presented in Fig. 12 were from a subject with a great deal of prior flight simulator experience. There was still a significant improvement in his operation when

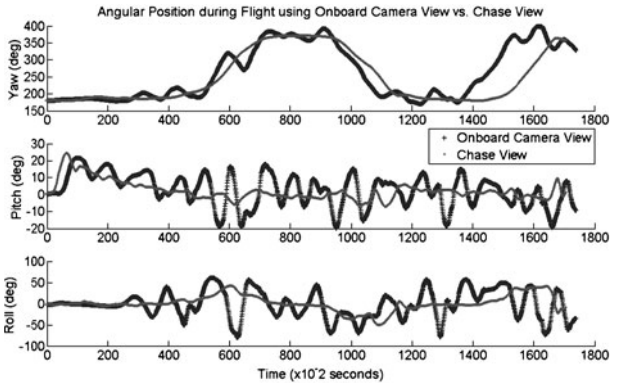

Fig. 12 Example data of the aircraft angular positions during an onboard camera and chase view test. The *thicker line* represents angles achieved using the onboard camera view the *thinner line* represents the angles achieved using the chase view

using the chase view over the onboard camera view. The increased field of view and virtual aircraft pose of the chase view decreased the need to oscillate back and forth to establish the aircraft's position in the environment. The surrounding view also helped with the subject's response to the camera motion lag and vibrations from SISTR as the virtual vehicle pose made it easier for the subjects to predict the desired angular position.

6 Conclusion and Future Works

6.1 Conclusions

Future applications for UAVs will take them into low flying areas populated with obstacles and civilians. Increased situational awareness for the pilots and operators controlling those UAVs will most certainly help decrease the potential for crashes and thereby decrease the chances of property damage or harm to civilians. This paper presented the development and evaluation of implementing a chase viewpoint for UAV operations. Results from the experiments show that the chase view method has potential to increase the situational awareness of UAV pilots. The results also showed that the chase view resulted in smoother motions and flight paths for the UAV.

6.2 Future Works

The chase view method can certainly use more validation. Future work involves testing current Predator Pilots and other UAV operators. These are subjects who are experts in UAV operation using onboard camera views. It will also be interesting to study if the chase view method will improve the speed of UAV pilot training. Real world field tests are also desired for complete validation of the chase view. Method I presented in this paper is being developed in parallel with Method II. Future work includes continued development of Method I and human tests for validation. Method I is desired for operations in environments where the terrain and obstacles are not known a priori. Method I as stated in this paper can also be adapted to handle areas with no GPS by obtaining vehicle state information from structure from motion algorithms. There may also be benefit to combining both Method I and II into a hybrid of both where Method I can be used to enhance areas where Method II fails such as when the flight environment changes from the model during flight. This will also be evaluated.

References

1. Endlesy, M.: Design and evaluation for situation awareness enhancements. In: Proceedings of the Human Factors Society 32nd Annual Meeting, pp. 97–101 (1988)
2. Oh, P.Y., Valavanis, K., Woods, R.: Uav workshop on civilian applications and commercial opportunities. (2008)
3. Weibel, R.E., Hansman, R.J.: Safety considerations for operation of unmanned aerial vehicles in the national airspace system. Tech. Rep. ICAT-2005-1, MIT International Center for Air Transportation (2005)
4. Defense, D.o.: Unmanned aircraft systems roadmap 2005-2030. Tech. rep. (2005)

5. Murphy, R.: Human-robot interaction in rescue robotics. IEEE Trans. Syst. Man Cybern. **34**(2), 138–153 (2004)
6. Hing, J.T., Oh, P.Y.: Development of an unmanned aerial vehicle piloting system with integrated motion cueing for training and pilot evaluation. J. Intell. Robot. Syst. **54**, 3–19 (2009)
7. Williams, K.W.: A summary of unmanned aircraft accident/incident data: Human factors implications. Tech. Rep. DOT/FAA/AM-04/24, US Department of Transportation Federal Aviation Administration, Office of Aerospace Medicine (2004)
8. Calhoun, G., Draper, M.H., Ruff, H.A., Fontejon, J.V.: Utility of a tactile display for cueing faults. In: Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting, pp. 2144–2148 (2002)
9. Ruff, H.A., Draper, M.H., Poole, M., Repperger, D.: Haptic feedback as a supplemental method of altering uav operators to the onset of turbulence. In: Proceedings of the IEA 2000/ HFES 2000 Congress, pp. 3.14–3.44 (2000)
10. Little, K.: Raytheon announces revolutionary new 'cockpit' for unmanned aircraft—an industry first (2006)
11. Tadema, J., Koeners, J., Theunissen, E.: Synthetic vision to augment sensor-based vision for remotely piloted vehicles. In: Enhanced and Synthetic Vision, vol. 6226, pp. 62260D–1–10. SPIE-Int. Soc. Opt. Eng. (2006)
12. Sugimoto, M., Kagotani, G., Nii, H., Shiroma, N., Matsuno, F., Inami, M.: Time follower's vision: a teleoperation interface with past images. IEEE Comput. Graph. Appl. **25**(1), 54–63 (2005)
13. Nielsen, C.W., Goodrich, M.A., Ricks, R.W.: Ecological interfaces for improving mobile robot teleoperation. IEEE Trans. Robot. **23**(5), 927–941 (2007)
14. Drury, J.L., Richer, J., Rackliffe, N., Goodrich, M.A.: Comparing situation awareness for two unmanned aerial vehicle human interface approaches. Tech. rep., Defense Technical Information Center OAI-PMH Repository. http://stinet.dtic.mil/oai/oai (United States) (2006)
15. Quigley, M., Goodrich, M. A., Beard, R.: Semi-autonomous human-uav interfaces for fixed-wing mini-uavs. 28 September–2 October 2004
16. Webb, T.P., Prazenica, R.J., Kurdila, A.J., Lind, R.: Vision-based state estimation for autonomous micro air vehicles. J. Guid. Control Dyn. **30**(3), 816–826 (2007)
17. Prazenica, R.J., Watkins, A.S., Kurdila, A.J., Ke, Q.F., Kandae, T.: Vision-based kalman filtering for aircraft state estimation and structure from motion. In: AIAA Guidance, Navigation, and Control Conference, vol. v 3, pp. 1748–1760. American Institute of Aeronautics and Astronautics, Reston (2005)
18. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600. IEEE, Piscataway (1994)
19. Bouguet, J.Y.: Pyramidal implementation of the lucas kanade feature tracker: description of the algorithm. Tech. rep., Intel Corporation Microprocessor Research Labs (2002)
20. Watkins, A.S., Kehoe, J.J., Lind, R.: Slam for flight through urban environments using dimensionality reduction. In: AIAA Guidance, Navigation, and Control Conference, vol. v 8, pp. 5018–5029. American Institute of Aeronautics and Astronautics, Reston (2006)
21. Narli, V., Oh, P.Y.: Hardware-in-the-loop test rig to capture aerial robot and sensor suite performance metrics, p. 2006. In: IEEE International Conference on Intelligent Robots and Systems (2006)
22. Meyer, A.: X-plane by laminar resarch. www.x-plane.com (2009)
23. Ernst, D., Valavanis, K., Garcia, R., Craighead, J.: Unmanned vehicle controller design, evaluation and implementation: from matlab to printed circuit board. J. Intell. Robot. Syst. **49**, 85–108 (2007)