A UAV Search and Rescue Scenario with Human Body Detection and Geolocalization*

Patrick Doherty and Piotr Rudol

Department of Computer and Information Science
Linköping University, SE-58183 Linköping, Sweden
{patdo,pioru}@ida.liu.se

Abstract. The use of Unmanned Aerial Vehicles (UAVs) which can operate autonomously in dynamic and complex operational environments is becoming increasingly more common. The UAVTech Lab [1], is pursuing a long term research endeavour related to the development of future aviation systems which try and push the envelope in terms of using and integrating high-level deliberative or AI functionality with traditional reactive and control components in autonomous UAV systems. In order to carry on such research, one requires challenging mission scenarios which force such integration and development. In this paper, one of these challenging emergency services mission scenarios is presented. It involves search and rescue for injured civilians by UAVs. In leg I of the mission, UAVs scan designated areas and try to identify injured civilians. In leg II of the mission, an attempt is made to deliver medical and other supplies to identified victims. We show how far we have come in implementing and executing such a challenging mission in realistic urban scenarios.

1 Introduction

The use of Unmanned Aerial Vehicles (UAVs) which can operate autonomously in dynamic and complex operational environments is becoming increasingly more common. While the application domains in which they are currently used are still predominantly military in nature, we can expect to see widespread usage in the civil and commercial sectors in the future as guidelines and regulations are developed by aeronautics authorities for insertion of UAVs in civil airspace.

One particularly important application domain where UAVs could be of great help in the future is in the area of catastrophe assistance. Such scenarios include natural disasters such as earthquakes or tsunamis or man-made disasters caused by terrorist activity. In such cases, civil authorities often require a means of acquiring an awareness of any situation at hand in real-time and the ability to monitor the progression of events in catastrophe situations. Unmanned aerial

* This work is supported in part by the National Aeronautics Research Program NFFP04 S4202 and the Swedish Foundation for Strategic Research (SSF) Strategic Research Center MOVIII and a Swedish Research Council grant.

[1] Autonomous Unmanned Aerial Vehicle Technologies Lab, Linköping University, Sweden, http://www.ida.liu.se/~patdo/auttek/

M.A. Orgun and J. Thornton (Eds.): AI 2007, LNAI 4830, pp. 1–13, 2007.

vehicles offer an ideal platform for acquiring the necessary situation awareness to proceed with rescue and relief in many such situations. It is also often the case that there is no alternative in acquiring the necessary information because one would like to avoid placing emergency services personal in the line of danger as much as possible.

For a number of years, The Autonomous Unmanned Aerial Vehicle Technologies Lab (UAVTech Lab) at Linköping University, Sweden, has pursued a long term research endeavour related to the development of future aviation systems in the form of autonomous unmanned aerial vehicles [1,2]. The focus has been on both high autonomy (AI related functionality), low level autonomy (traditional control and avionics systems), and their integration in distributed software architectural frameworks [3] which support robust autonomous operation in complex operational environments such as those one would face in catastrophe situations.

More recently, our research has moved from single platform scenarios to multi-platform scenarios where a combination of UAV platforms with different capabilities are used together with human operators in a mixed-initiative context with adjustable platform autonomy. The application domain we have chosen to pursue is emergency services assistance. Such scenarios require a great deal of cooperation among the UAV platforms and between the UAV platforms and human operators.

The paper is structured in the following manner. In section 2, we introduce the emergency services scenario. In section 3, we describe the UAV platforms used in the scenario. In section 4, we consider the body identification and geolocation phase of the mission in more detail and in section 5, we consider the supply delivery phase of the mission in more detail.

2 An Emergency Service Scenario

On December 26, 2004, a devastating earthquake of high magnitude occured off the west coast off Sumatra. This resulted in a tsunami which hit the coasts of India, Sri Lanka, Thailand, Indonesia and many other islands. Both the earthquake and the tsunami caused great devastation. During the initial stages of the catastrophe, there was a great deal of confusion and chaos in setting into motion rescue operations in such wide geographic areas. The problem was exacerbated by shortage of manpower, supplies and machinery. Highest priorities in the initial stages of the disaster were search for survivors in many isolated areas where road systems had become inaccessible and providing relief in the form of delivery of food, water and medical supplies.

Let's assume for a particular geographic area, one had a shortage of trained helicopter and fixed-wing pilots and/or a shortage of helicopters and other aircraft. Let's also assume that one did have access to a fleet of autonomous unmanned helicopter systems with ground operation facilities. How could such a resource be used in the real-life scenario described?

A pre-requisite for the successful operation of this fleet would be the existence of a multi-agent (UAV platforms, ground operators, etc.) software infrastructure for assisting emergency services in such a catastrophe situation. At the very least, one would require the system to allow mixed initiative interaction with multiple platforms and ground operators in a robust, safe and dependable manner. As far as the individual platforms are concerned, one would require a number of different capabilities, not necessarily shared by each individual platform, but by the fleet in total. These capabilities would include:

– the ability to scan and search for salient entities such as injured humans, building structures or vehicles;
– the ability to monitor or surveil these salient points of interest and continually collect and communicate information back to ground operators and other platforms to keep them situationally aware of current conditions;
– the ability to deliver supplies or resources to these salient points of interest if required. For example, identified injured persons should immediately receive a relief package containing food, medical and water supplies.

Although quite an ambitious set of capabilities, several of them have already been achieved to some extent using our experimental helicopter platforms, although one has a long way to go in terms of an integrated, safe and robust system of systems.

To be more specific in terms of the scenario, we can assume there are two separate legs or parts to the emergency relief scenario in the context sketched previously.

Leg I. In the first part of the scenario, it is essential that for specific geographic areas, the UAV platforms should cooperatively scan large regions in an attempt to identify injured persons. The result of such a cooperative scan would be a saliency map pinpointing potential victims, their geographical coordinates and sensory output such as high resolution photos and thermal images of potential victims. The resulting saliency map would be generated as the output of such a cooperative UAV mission and could be used directly by emergency services or passed on to other UAVs as a basis for additional tasks.

Leg II. In the second part of the scenario, the saliency map generated in Leg I would be used as a basis for generating a logistics plan for several of the UAVS with the appropriate capabilities to deliver food, water and medical supplies to the injured identified in Leg I. This of course would also be done in a cooperative manner among the platforms.

3 Hardware Platform

The UAVTech UAV platform [1] is a slightly modified Yamaha RMAX helicopter (Fig. 1). It has a total length of 3.6 m (including main rotor) and is powered by a 21hp two-stroke engine with a maximum takeoff weight of 95 kg. The on-board

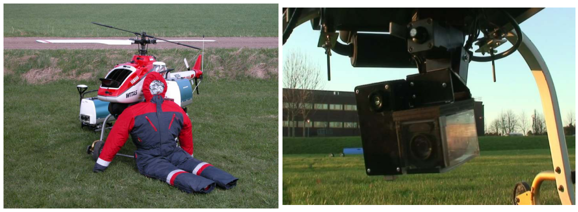

Fig. 1. The UAVTech UAV and the on-board camera system mounted on a pan-tilt unit

system contains three PC104 embedded computers. The primary flight control (PFC) system includes a Pentium III 700Mhz, a wireless Ethernet bridge, a GPS receiver, and several additional sensors including a barometric altitude sensor. The PFC is connected to the RMAX helicopter through the Yamaha Attitude Sensor (YAS) and Yamaha Attitude Control System (YACS), an image processing computer and a computer responsible for deliberative capabilities. The deliberative/reactive system (DRC) runs on the second PC104 embedded computer (Pentium-M 1.4GHz) and executes all high-end autonomous functionalities such as mission or path planning. Network communication between computers is physically realized with serial lines RS232C and Ethernet.

The image processing system (IPC) runs on the third PC104 embedded Pentium III 700MHz computer. The camera platform suspended under the UAV fuselage is vibration isolated by a system of springs. The platform consists of a Sony CCD block camera FCB-780P and a ThermalEye-3600AS miniature infrared camera mounted rigidly on a Pan-Tilt Unit (PTU) as presented in Fig. 1. The video footage from both cameras is recorded at a full frame rate by two miniDV recorders to allow processing after a flight.

4 Mission Leg I: Body Identification

The task of the 1st leg of the mission is to scan a large region with one or more UAVs, identify injured civilians and output a saliency map which can be used by emergency services or other UAVs. The technique presented uses two video sources (thermal and color) and allows for high rate human detection at larger distances then in the case of using the video sources separately with standard techniques. The high processing rate is essential in case of video collected onboard a UAV in order not to miss potential objects as a UAV flies over it. A thermal image is analyzed first to find human body sized silhouettes. Corresponding regions in a color image are subjected to a human body classifier which is configured to allow weak classifications. This focus of attention allows for maintaining a body classification at a rate up to 25Hz. This high processing rate allows for collecting statistics about classified humans and to prune false classifications of the "weak" human body classifier. Detected human bodies are geolocalized on a map which can be used to plan supply delivery. The technique presented has been tested on-board the UAVTech helicopter platform and is an important component in our research with autonomous search and rescue missions.

4.1 Image Processing

Video footage collected by a UAV differs substantially from images acquired on the ground and the use of standard techniques is not straight forward. For instance, both maximum and minimum speeds are determined by an aircraft's properties. Nevertheless, high flight speed is preferred in case of search and rescue applications. Therefore it is essential for the image processing algorithm to perform close to the full frame rate to process all frames of the video.

The algorithm we use takes as input two images (camera planes are assumed to be close to parallel to the earth plane) and the processing starts by analyzing the thermal image. The image is first thresholded to find regions of human body temperature. The shape of the regions is analyzed and those which do not resemble a human body (i.e. wrong ratio of minor and major axes of the fitted ellipse and incorrect area) are rejected. Additionally, regions which lie on the image border are rejected as they may belong to a bigger warm object. Once human body candidates are found in the thermal image, corresponding regions in the color image are calculated.

Computation of the corresponding region in the color image could be achieved by performing image registration or feature matching in both images. The former technique is too time consuming and the latter is infeasible because of mostly different appearance of features in color and thermal images. Here, a closed form solution is used which takes into account information about the UAV's state.

Computation of the corresponding region in the color image starts with calculating coordinates of a point T ($\tilde{v_T}$) whose projection is the pixel in the thermal image $\tilde{u_t}$ i.e.

$$\tilde{u_t} = P_t \tilde{v_T} \quad \tilde{u_t} \in \mathcal{P}^2 \quad \tilde{v_T} \in \mathcal{P}^3 \tag{1}$$

where P_t represents extrinsic and intrinsic parameters of the thermal camera. The general scheme of the problem is shown in Figure 2. A line equation with the direction vector $\tilde{v}_{cam}$ which goes through camera center through pixel $\tilde{u_t}$ and intersects the ground plane in point T is:

$$\tilde{v}_T - \tilde{v}_C = t \cdot \tilde{v}_{cam} \quad t \in \mathbb{R} \tag{2}$$

The ground plane is defined by the point G($\tilde{v}_G$) and the normal vector $\tilde{n}$ which is the down component of the NED (North, East, Down) frame:

$$(\tilde{v}_T - \tilde{v}_G) \cdot \tilde{n} = 0 \tag{3}$$

Finally, the vector $\tilde{v}_T$ which describes the point of intersection of a ray of light going through the camera center and the pixel of the target can be calculated according to:

$$\tilde{v}_T = \tilde{v}_C + \frac{(\tilde{v}_G - \tilde{v}_C) \cdot \tilde{n}}{\tilde{v}_{cam} \cdot \tilde{n}} \cdot \tilde{v}_{cam} \tag{4}$$

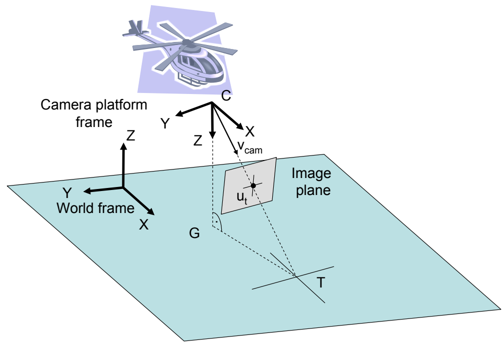

Fig. 2. Calculation of a target coordinates

In order to calculate $\tilde{v}_{cam}$ the vector along the X axis of the camera frame must be expressed in the world coordinate frame. This transformation can be expressed as:

$$^{w}\tilde{v}_{cam} = P_{heli} P_{ptu} P_p \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T \tag{5}$$

where P_p describes the transformation depending on the undistorted pixel position $\tilde{u_t}$. Matrix P_{ptu} is built to represent a transformation introduced by the pan-tilt unit. P_{heli} represents the attitude of the UAV and is built up from roll, pitch and yaw angles delivered by the YAS system.

The method presented can be extended to relax the flat world assumption. The point T can be found by performing ray-tracing along the line described by equation Eq. 2 to find the intersection with the ground elevation map.

Calculated world position can additionally be checked against the on-board geographic information database to verify whether the calculated point is valid. Depending on the situation, certain positions can be excluded from the map. If the world position is accepted, its projection is calculated for the color camera using the following formula:

$$\tilde{u}_c = P_c \tilde{v}_T \quad \tilde{u}_c \in \mathcal{P}^2 \quad \tilde{v} \in \mathcal{P}^3 \tag{6}$$

where P_c constitutes the matrix encoding intrinsic and extrinsic parameters of the color camera.

4.2 The Classifier

Once the corresponding pixel in the color image is identified, a sub-window with the pixel $\tilde{u}_c$ in the center is selected and it is subjected to an object detector first suggested by [11]. The work was a basis for several improvements, one of which was presented in [9]. One of these included extending the original feature set which is presented in Fig. 3.

The classifier which is in fact a cascade of boosted classifiers working with Haar-like features requires training with a few hundred positive and negative examples. During learning the structure of a classifier is learned using boosting.

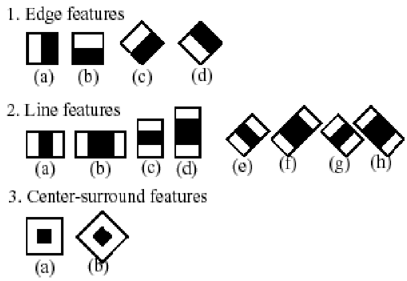

Fig. 3. Leinhart's extended set of available features

The use of a cascade of classifiers allows for dramatic speed up of computations by skipping negative instances and only computing features with high probability for positive classification. The speed up comes from the fact that the classifier, as it slides a window at all scales, works in stages and is applied to a region of interest until at some stage the candidate is rejected or all the stages are passed. This way, the classifier quickly rejects subregions which most probably do not include features needed for positive classification (i.e. background processing is quickly terminated). The classifier works with features which can be quickly extracted using intermediate image representations - integral images. The reason for working with features instead of pixel intensities is that features encode knowledge about the domain, which is difficult to learn from raw input data. The features encode the existence of oriented contrasts between regions of an image. The Haar-like features used here can be calculated at any position and any scale in constant time using only eight look-ups in the integral image.

The classifier used in this work is a part of the Open Source Computer Vision Library [10] and the trained classifier for upper-, lower- and full human body is a result of [8]. The trained classifier is best suited for pedestrian detection in frontal and backside views which is exactly the type of views a UAV has when flying above the bodies lying on the ground.

Since the body classifier is configured to be "relaxed" it delivers sporadic false positive classifications. To counter for most of them the following method is used to prune the results. Every salient point in the map has two parameters which are used to calculate certainty of a location being a human body: T_{frame} which describes the amount of time a certain location was in the camera view and T_{body} which describes the amount of time a certain location was classified as a human body. The certainty factor is calculated as follows:

$$p_{body}(loc_i) = \frac{T_{body}}{T_{frame}} \tag{7}$$

A location is considered a body if $p_{body}(loc_i)$ is larger than a certain threshold (e.g. 0.5 during the flight tests) and T_{frame} is larger than a desired minimal observation time. Locations are considered equal if geographical distance between them is smaller then a certain threshold (depending on the geolocation accuracy) and the final value of a geolocalized position is an average of the observations (c.f. Section 4.4).

4.3 Experimental Setup

A series of flight tests were performed in southern Sweden at an emergency services training center used by the Swedish Rescue Services Agency to train fire, police and medical personnel. Flight tests were performed over varied terrain such as asphalt and gravel roads, grass, trees, water and building roof tops which resulted in a variety of textures in the images. Two UAVs were used over a search area of 290x185 meters. A total of eleven bodies (both human and dummies with close to human temperature) were placed in the area. The goal of the mission was to generate a saliency map. The general mission plan is shown in Fig. 4. Before take-off, one of the UAVs was given an area to scan (dashed

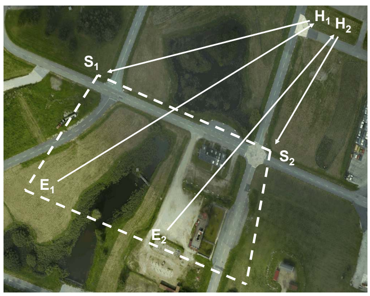

Fig. 4. Mission overview

line polygon). It then delegated part of the scanning task to another platform, generating sub-plans for itself and the other platform. The mission started with a simultaneous autonomous take-off at positions H_1 and H_2 and the UAVs flew to starting positions S_1 and S_2 for scanning. Throughout the flights, saliency maps were incrementally constructed until the UAVs reached their ending positions E_1 and E_2. The UAVs then returned to their respective take-off positions for a simultaneous landing. The mission took approximately ten minutes to complete and each UAV traveled a distance of around 1km.

4.4 Experimental Results

The algorithm found all eleven bodies placed in the area. The saliency map generated by one of the helicopters is shown in Fig. 5. The images of identified objects are presented in Fig. 6. Several positions were rejected as they were not observed long enough (i.e. 5 seconds). Images 7, 9, and 14 present three falsely identified objects.

The accuracy of the body geolocation calculation was performed by measuring GPS (without differential correction) positions of bodies after an experimental

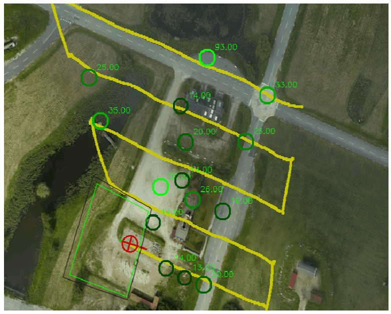

Fig. 5. Flight path and geolocated body positions

Fig. 6. Images of classified bodies. Corresponding thermal images are placed under color images.

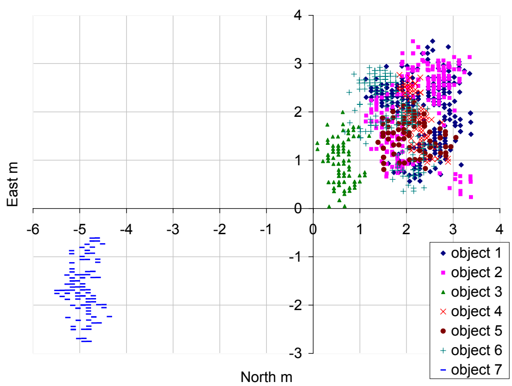

Fig. 7. Geolocation error for multiple objects

flight. The accuracy of the system is sufficient for the application of delivering supplies to the detected humans. Figure 7 presents the error measurement for seven geolocated objects. The measurement has a bias of approximately two meters in both east and north directions. It is a sum of errors in GPS measurement, accuracy of the camera platform mounting, PTU measurement, and camera calibration inaccuracies. The spread of measurement samples of approximately 2.5 meters in both east and north directions is caused by the error of attitude measurement, the system of springs in the camera platform, the flat ground assumption, and time differences between UAV state estimate, PTU angle measurement and image processing result acquisition. A large geolocation error of object 7 is caused by erroneous GPS measurement. Object 7 was located on a metal foot-bridge and the GPS antenna during static measurement was additionally partially occluded by metal railings. The noise on the measurement however is consistent with the rest of the objects.

5 Mission Leg II: Package Delivery

After successful completion of leg I of the mission scenario, we can assume that a saliency map has been generated with geo-located positions of the injured civilians. In the next phase of the mission, the goal is to deliver configurations of medical, food and water supplies to the injured. In order to achieve this leg of the mission, one would require a task planner to plan for logistics, a motion planner to get one or more UAVS to supply and delivery points and an execution monitor to monitor the execution of highly complex plan operators. Each of these functionalities would also have to be tightly integrated in the system. These components are described in section 5.1

Currently, we have developed this mission leg primarily in simulation with hardware-in-the-loop. Our avionics boxes are coupled directly to a simulator and execute all functionalities necessary for completion of the mission in the actual hardware we fly missions with. A physical winch system for picking up and putting down packages is currently under development.

For these logistics missions, we assume the use of one or more UAVs with diverse roles and capabilities. Initially, we assume there are n injured body locations, several supply depots and several supply carrier depots (see figure 8).

5.1 Planning, Execution and Monitoring

Figure 9 shows part of our UAV system architecture, with an emphasis on those components that are the most relevant for planning, execution, and execution monitoring.

At the top of the center column is the *plan executor* which given a mission request, calls *DyKnow* [6,7], a knowledge processing middleware, to acquire essential information about the current contextual state of the world or the UAV's own internal states. Together with a domain specification and a goal specification related to the logistics scenario, this information is fed to *TALplanner* [4,5], a logic-based task planner which outputs a plan that will achieve the designated goals, under the assumption that all actions succeed and no failures occur. Such a plan can also be automatically annotated with global and/or operator-specific

Fig. 8. A Supply Depot (left) and a Carrier Depot (right)

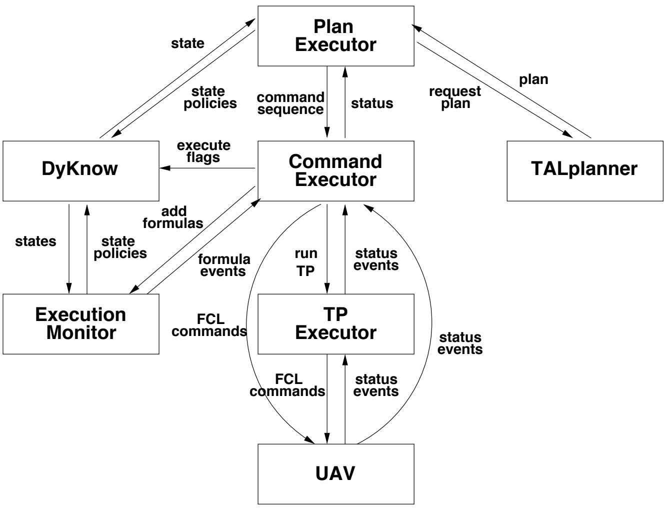

Fig. 9. System Architecture Overview

conditions to be monitored during execution of the plan by an execution monitor in order to relax the assumption that no failures can occur. Such conditions are expressed as temporal logical formulas and evaluated on-line using formula progression techniques. This execution monitor notifies the plan executor when actions do not achieve their desired results and one can then move into a plan repair phase.

The plan executor translates operators in the high-level plan returned by TALplanner into lower level command sequences which are given to the *command executor*. The command executor is responsible for controlling the UAV, either by directly calling the functionality exposed by its lowest level Flight Command

Language (FCL) interface or by using so called *task procedures* (TPs) through the *TP Executor* subsystem. The TP Executor is part of the Modular Task Architecture (MTA) [3], which is a reactive system designed in the procedure-based paradigm and developed for loosely coupled heterogeneous systems. A task is a behavior intended to achieve a goal in a limited set of circumstances. A task procedure is the computational mechanism that achieves this behavior. The TPs have the ability to use deliberative services, such as the task planner described above or motion planners [12,13], in a reactive or contingent manner and to use traditional control services in a reactive or contingent manner and thereby integrate deliberation and reaction.

During plan execution, the command executor adds formulas to be monitored to the *execution monitor*. DyKnow continuously sends information about the development of the world in terms of state sequences to the monitor, which uses a progression algorithm to partially evaluate monitor formulas. If a violation is detected, this is immediately signaled as an event to the command executor, which can suspend the execution of the current plan, invoke an emergency brake command, optionally execute an initial recovery action, and finally signal new status to the plan executor. The plan executor is then responsible for completing the recovery procedure.

The fully integrated system is implemented on our UAVs and can be used onboard for different configurations of the logistics mission described in Leg II of the larger mission. The simulated environments used are in urban areas and quite complex. Plans are generated in the millisecond to seconds range using TALplanner and empirical testing shows that this approache is promising in terms of integrating high-level deliberative capability with lower-level reactive and control functionality.

6 Conclusions

We have described a realistic emergency services scenario and shown how far we have come in the deployment of autonomous UAV systems which require the use of deliberative, reactive and control capabilities in a highly integrated and time-constrained context. Currently, we are developing a winch system for the RMAX which will be used to deliver supplies of the type described in leg II of the scenario. We are also refining the body identification algorithms and developing a framework for cooperation based on the use of delegation of goals and action sequences.

References

1. Doherty, P.: Advanced research with autonomous unmanned aerial vehicles. In: Proceedings on the 9th International Conference on Principles of Knowledge Representation and Reasoning (2004)
2. Doherty, P.: Knowledge representation and unmanned aerial vehicles. In: IAT 2005. Proceedings of the IEEE Conference on Intelligent Agent Technolology (2005)

3. Doherty, P., Haslum, P., Heintz, F., Merz, T., Persson, T., Wingman, B.: A Distributed Architecture for Autonomous Unmanned Aerial Vehicle Experimentation. In: Proceedings of the International Symposium on Distributed Autonomous Robotic Systems, pp. 221–230 (2004)
4. Doherty, P., Kvarnström, J.: TALplanner: A temporal logic based planner. Artificial Intelligence Magazine (Fall Issue, 2001)
5. Doherty, P., Kvarnström, J.: Temporal action logics. In: Lifschitz, V., van Harmelen, F., Porter, F. (eds.) The Handbook of Knowledge Representation, vol. ch.18, Elsevier, Amsterdam (to appear, 2007)
6. Heintz, F., Doherty, P.: Dyknow: An approach to middleware for knowledge processing. Journal of Intelligent and Fuzzy Systems 15(1), 3–13 (2004)
7. Heintz, F., Doherty, P.: Dyknow: A knowledge processing middleware framework and its relation to the JDL fusion model. Journal of Intelligent and Fuzzy Systems 17(4) (2006)
8. Kruppa, H., Castrillon-Santana, M., Schiele, B.: Fast and robust face finding via local context. In: Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (October 2003)
9. Lienhart, R., Maydt, J.: An Extended Set of Haar-like Features for Rapid Object Detection. In: Proc. of International Conference on Image Processing, pp. 900–903 (2002)
10. OSCVL. Open Source Computer Vision Library, `http://www.intel.com/technology/computing/opencv/`
11. Viola, P., Jones, M.J.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Proc. of Conference on Computer Vision and Pattern Recognition (2001)
12. Wzorek, M., Conte, G., Rudol, P., Merz, T., Duranti, S., Doherty, P.: From Motion Planning to Control - A Navigation Framework for an Autonomous Unmanned Aerial Vehicle. In: Proc. of the 21th Bristol International UAV Systems Conference (2006)
13. Wzorek, M., Doherty, P.: A framework for reconfigurable path planning for autonomous unmanned aerial vehicles. Journal of Applied Artificial Intelligence (2007)