International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XXXVIII, Part 5
Commission V Symposium, Newcastle upon Tyne, UK. 2010

A LOW-COST AIRBORNE PLATFORM FOR ECOLOGICAL MONITORING

A. F. Clark,* J. C. Woods and O. Oechsle

School of Computer Science and Electronic Engineering, University of Essex
Wivenhoe Park, Colchester CO4 3SQ, UK
{alien,woodjt,ooechs}@essex.ac.uk

Commission V

KEY WORDS: UAV, autonomous aerial robot, ecological monitoring, genetic programming

ABSTRACT:

This work documents the development of an aerial environmental monitoring platform based on a paramotor, dubbed *robofoil*. Significant advantages are achieved in safety, durability, ease of use and flexibility by employing an inflated wing. The aircraft is easy to fly, has near vertical ascent into wind and an intrinsic fail-safe. The ability to control the wing angle of attack and interchange wings according to weather or mission requirements makes this platform truly flexible. With an onboard autopilot and manual override, the vehicle is intuitive to fly and has a short learning curve for the user. With flight speeds ranging from 0 to 40 knots, the vehicle is well-suited to targeted surveillance as well as being resilient to gusty conditions. With a high payload capability, the platform can carry fuel for flights in excess of an hour in the current version. We have established that it is possible to use genetic programming, a machine learning technique, to evolve application-specific systems purely through training. Our eventual aim is for the design, construction details and software used for robofoil to be made fully open.

1. INTRODUCTION

Satellite remote sensing is now able to deliver imagery with stunning ground resolution, captured using a wide range of wavebands and modalities. While access to such imagery is fairly straightforward for governments and agencies, for groups or individuals acquiring imagery of particular regions taken at particular times remains costly, usually prohibitively so. Moreover, there remain some technical shortcomings: images cannot be captured while there is cloud cover (at least in visible wavebands), satellites may not pass over a region at the right time, and there is difficulty in capturing 3D structure. These shortcomings mitigate towards the use of a complementary capture approach, and there has been a significant amount of recent research into unmanned aerial vehicles (UAVs), which hold much promise (Zhou *et al*, 2009).

The research described here is motivated by the desire to produce a low-cost UAV that can be used for ecological monitoring in remote regions, particularly the rural Amazon. It is a report on work in progress, and the focus is on producing "proof of concept" systems at this juncture, evolving a more mature system in the light of experience. The need for the UAV to be flown in remote regions means it must be robust, yet easily repaired and maintained using widely-available components. These *desiderata* and how well existing UAV designs meet them are discussed in Section 2. This leads on to a description of our solution in Sections 3 and 4. Section 5 discusses applications we have explored to date, and Section 6 presents some concluding remarks and outlines further work.

2. UAV *DESIDERATA*

As indicated above, the aim of this work is to produce an airborne platform, sensors, and associated processing that can be used in fairly remote regions. The target applications are principally in the general area of ecological monitoring but it is intended that the platform will support other remote sensing applications too; one of these is described in Section 5.

Many ecological monitoring applications involve data capture that is regular and frequent over extended periods of time. As small UAVs tend not to fly well in heavy rain or strong winds, it is important that the UAV can be deployed quickly to take advantage of short breaks in otherwise bad weather. Furthermore, ecological monitoring typically involves only one or two people, so the UAV needs to be able to be used by an individual, including launching – this is in contrast to the three-man team involved in (Gay *et al*, 2009) to meet the CAA guidelines in the UK (CAA, 2009).

Monitoring tasks usually involve observing a reasonable area – the size of the region we expect to monitor is roughly 5 x 3 km. To obtain the best quality imagery, it is desirable to have a low airspeed, which means the UAV needs to generate a fair amount of lift.

It is likely that different monitoring tasks will involve different types of sensing, so the UAV platform and its operation need to independent of its payload. Good quality compact digital still and movie cameras are adequate for most data capture in the visible waveband and typically weigh about 0.25 kg; however, small infra-red, hyperspectral and radar sensors are all somewhat heavier, where available, so it is desirable for the UAV to be able to carry a payload of several kilograms.

* Corresponding author

Even with experienced operators, the attrition rate for UAVs is high. There are two consequences of this: firstly, as much of the airframe as possible must be resistant to impacts; and secondly, it must be possible to obtain spares and perform repairs in the field quickly. The latter in particular implies a need for good documentation and easy-to-obtain components; of course, even very rural areas generally have good machine shops for the repair of agricultural equipment.

Having outlined the *desiderata* for a UAV, let us consider how well existing designs meet them. Commercial UAVs fall into two broad categories. Those that originate from military technologies tend to have large airframes, are able to carry significant payloads...but are far too expensive for the kind of system envisaged here. At the other extreme, UAVs based around radio-controlled model aircraft are low-cost and easily available – so, like many other researchers, this is the most practical route for this work.

Radio-controlled aircraft are normally helicopters or have fixed-wing airframes. Miniature helicopters (say, under 20 cm blade size) are unable to carry a payload of more than a few tens of grams; this makes them fine for carrying miniature cameras but for little else (De Nardi *et al*, 2006). Being so light, they can be difficult to keep stable indoors and almost impossible to fly successfully outdoors. Larger single-blade helicopters are normally powered by petrol engines and are able to carry an adequate payload – but the rotor is easily capable of removing fingers and even limbs and require skilled pilots, so we have discarded this solution.

Between these two extremes lie quad-rotor helicopters, which have become popular with robotics researchers around the world. These are much more stable than smaller helicopters and rotor injuries are reduced to minor cuts. Commercially-available solutions are able to be flown outdoors; however, they are normally battery-powered and this constrains flight time – our experience is that ageing of the batteries quickly reduces the time they can stay aloft to be too short for our purposes. Furthermore, our experimental work with these devices shows that vibration from the motors causes noticeable blurring of captured images, so much so that we had to replace them with brushless motors.

Fixed-wing aircraft are by far the most commonly-used type of UAV, resulting from the ready availability of radio-controlled model kits. The comparative similarity to real piloted aircraft, their relative stability, and the ability to carry a reasonable payload make them an attractive option. Weighed against that, however, is the fact that even fairly minor crashes cause significant damage to the airframe and a skilled pilot is required, as for single rotor helicopters.

Largely because of this possibility of damage from minor crashes or poor landings, we have been investigating an alternative approach, which we have dubbed *robofoil* (Figure 1). Like (Dunford *et al*, 2009), we have taken microlight aircraft as our inspiration and used a paraglider/paramotor as the basis of our UAV design. The piloting skills required to fly this type of aircraft are low: the aircraft is difficult to stall, the flight envelope is heavily damped and the whole system has a forgiving nature. Take-off is hand-launched (Figure 2) or from a very short runway, depending on size, and into-wind ascents can be near vertical. In the event of pilot error, simply closing off the throttle causes the aircraft to glide back to the ground without significant risk to the ground-crew or innocent bystanders.

Figure 1. Robofoil in flight

Figure 2. Launching robofoil

Most important in the context of ecological monitoring in remote regions, however, is the low cost of the components and the simplicity of construction and repair. Moreover, the vehicle is compact: the wing rolls up and the fuselage stows in a modest-sized box, allowing easy and safe transportation.

As the system matures, it is our aim to make complete design and construction details for the complete system – not just the airframe but also control and processing software – available on the web. These are intended to be complete enough for anyone with reasonable mechanical and computing skills to be able to build and fly this design of UAV. Of course, there are other initiatives in place to make freely available at least some UAV components (OpenPilot, 2010).

3. ROBOFOIL: A KITE-BASED UAV

Rather than a conventional wing, paramotors use a fabric structure that resembles a parachute. As robofoil will carry a payload weighing much less than a man, a much smaller 'wing' can be used – we are actually using kites made by Flexifoil. Work to date has centred on Flexifoil 6, 8 and 10 kites. The Flexifoil 10 provides a surface area of about 1.75 m^2 and has a leading edge stiffened with a carbon fibre rod. The parafoil inflates when air passes through it and provides the cross-section required for lift.

An attraction of this approach is that different payloads can be accommodated by changing the kite – if the sensors being carried would compromise performance or a slow flight speed is required for a particular observation, the kite can be replaced with a larger one that generates more lift; it takes only a few seconds to undo the fast-release clips on one kite and attach another. Contingency on engine power is assumed and the fuselage is overpowered with the largest kite in mind. Interchangeable wings provide a degree of flexibility not enjoyed by fixed-wing solutions. In the event of ground impact, the wing simply collapses; and it is relatively immune to tearing from sharp objects because it is made of Ripstop spinnaker nylon.

The fuselage is suspended from the kite by lines which come from the leading edge and are secured to the steering mechanism, described below. In the same way that it is possible to change kites, it is possible to change airframes, so that a somewhat smaller gondola can be flown if, say, only video is required. This provides further flexibility and is also being used to ascertain the best system configurations for different applications. The basic chassis of the airframe is made from lengths of stiff wire (*e.g.*, welding rod) and the engine *etc* are attached to that (Figure 3).

The UAV is powered by an engine intended for a conventional radio-controlled model aircraft. We are experimenting with several engines, the largest of which is an OS Surpass-120, a miniature 4-stroke engine which can generate up to 2 hp; the smallest is an OS20 two-stroke producing a meagre 0.1 hp. The vehicle shown in Figure 3 drives a rear-mounted propeller – this reduces a little the likelihood of damage in crashes, though some care is needed when launching it. Whether to use a 'puller' (*i.e.*, the engine goes first) or a 'pusher' (the engine goes last) is a matter of some debate: each has its own advantages and disadvantages. For example the pusher shown in Figure 3 is easier to steer, but is a potential hazard for the wing bridles, whereas another version we use which is a puller type is more difficult to balance into level flight. Airframes subsequent to that shown in Figure 3 have the propeller encased in a light wire mesh for safety.

Steering is achieved simply by pulling down or pushing up a pivoted metal bar, in much the same way that a paraglider pilot steers by shifting his or her weight. Although there are no conventional control surfaces, this is entirely adequate for the kinds of manoeuvre that the UAV needs to perform. In our basic design, the trailing edge is rigged with bridles and secured to a servo that controls the angle of attack (Figure 4) – though we have recently been experimenting with a configuration that discards the latter, with promising results. The flight characteristics and the lift generated can be accurately controlled by altering this angle of attack servo. Flight speeds from near stationary to 40 knots can be achieved with the current version. The ability to control airspeed ultimately leads to a fail-safe mode where the wing is trimmed with a high angle of attack and the airspeed is minimal. Indeed one of the principal advantages of using a parafoil is safety: in the event of control loss, the aircraft simply behaves as a weight under a parachute would and floats harmlessly to the ground. The design of the wing is not critical and most parafoil-based kites, once correctly rigged, can be used. In the event of damage their replacement is cheap and easily sourced.

Fuel loads sufficient for flight durations in excess of an hour are currently carried, though these could be make considerably longer if required. This is something of a luxury: time in the air is generally severely restricted in micro-UAVs, both rotary- and fixed-wing.

The flight of the aircraft is currently controlled by a Micropilot MP2028 autopilot. This is a single circuit board, weighing only about 30 g, which provides position sensing via GPS and the control of flight surfaces; it is powered from a dedicated battery. It can be controlled using a conventional radio controller (we use a Spektrum DX7, which operates at 2.4 GHz) but with the ability to have waypoints uploaded pre-flight and then fly between them autonomously; switching from manual to autonomous flight is achieved through the radio controller once the UAV is in the air. Although this autopilot makes the construction of the UAV as a system fairly straightforward, it is much too expensive to be used in a *low-cost* UAV; moreover, it is available from only one manufacturer and requires an export licence, which contravenes our wish for easy maintenance in the field. Hence, as robofoil develops, the authors anticipate replacing the autopilot with a set of off-the-shelf components. To that end, we have extensive experience with Gumstix ARM boards, which we use to control miniature helicopters (De Nardi, 2006), with all the Kalman filtering running on the processor.

The autopilot software expects the normal control surfaces of a fixed-wing aircraft, which are obviously not available on a kite. These controls are mapped onto pulls and pushes of the steering bar. Although this works well enough, it is less elegant than the authors would like, so it anticipated that a customised control algorithm will be put in place when the autopilot is replaced.

4. ROBOFOIL SENSING AND PROCESSING

The whole point of the aerial platform, of course, is to provide a platform for sensing. In satellite remote sensing, one typically employs a high-quality, downward-looking camera to form what is essentially a map of the ground. This carries across to the UAV case, where compact digital cameras work well: they are fairly robust, light in weight, yet capture good quality images. The main *desiderata* here are RAW image capture, the ability to set the white balance, and of course an externally-controllable shutter.

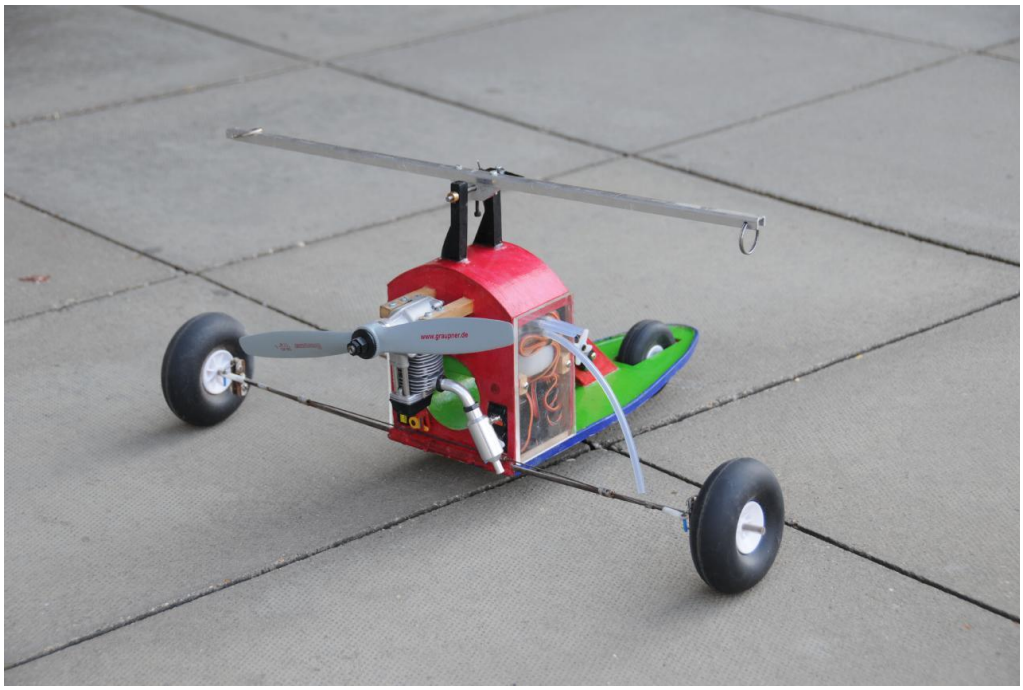

Figure 3. Robofoil

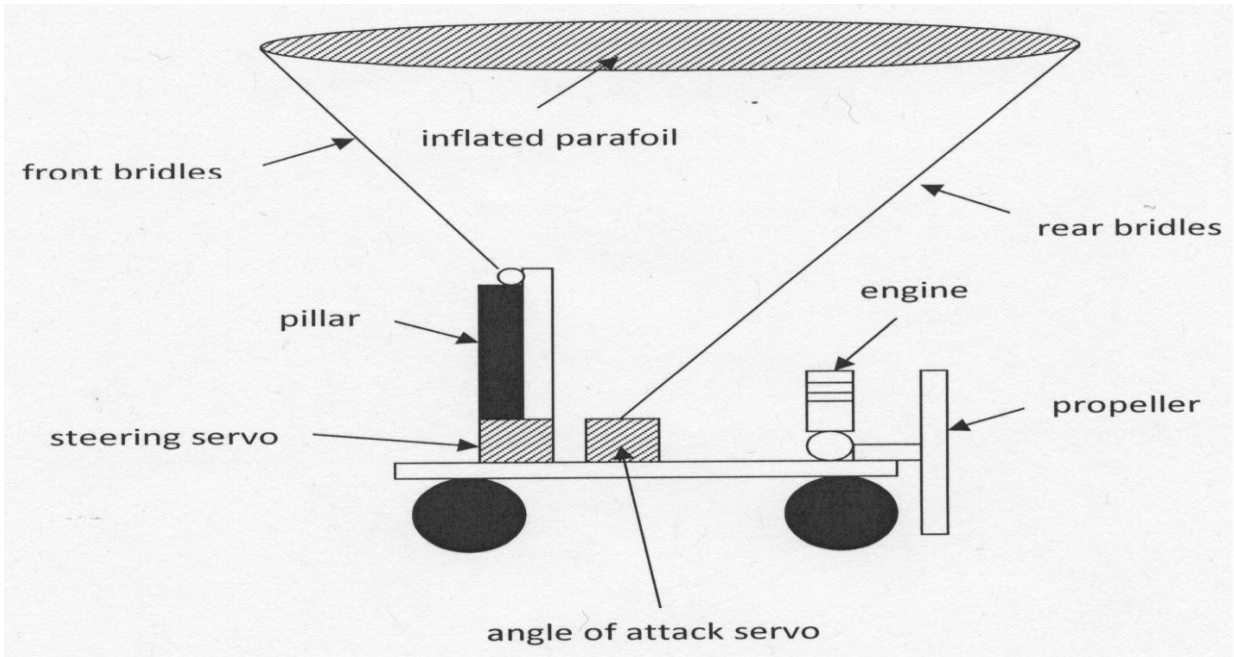

Figure 4. Adjusting the wing's angle of attack

It is conventional to 'stitch' together individual frames captured in this way into a mosaic. Fortunately, the computer vision community has devised a number of algorithms that are able to help automate this, and the current favourite is the *Scale-Invariant Feature Transform* (SIFT) (Lowe, 2004). SIFT detects 'good' feature points in a way that is reasonably scale- and rotation-independent, then identifies each one with a descriptor; these descriptors are normally saved in a file. To match a pair of partially-overlapped images, one uses SIFT on them, then searches for features with similar descriptors to identify match-points; one takes a consistent subset of these (found *e.g.*, using RANSAC or a Hough transform) to calculate the homography (transformation) between images.

Furthermore, there are some shortcomings with SIFT for processes other then mosaicking. When flying over a 3D

object, its appearance in a series of images undergoes an *out-of-plane* rotation, which SIFT is not invariant to – meaning that, in extreme cases, it will fail to match descriptors. Moreover, SIFT tends to avoid locating features on the boundaries of image features as their descriptors will differ depending on the background – but boundaries are normally precisely what one is interested in! Indeed, several vision researchers are using conventional edge- and corner-detectors (*e.g.*, Canny and Harris-Stephens) for boundaries and SIFT for texture within regions.

SIFT should be able to match feature points on 3D objects providing the out-of-plane rotation due to the motion of the aerial vehicle is fairly small; the best way to achieve this is to capture not a series of still images but a video sequence. To that end, the authors are experimenting with small video cameras on robofoil. The device currently being used is a FlyCamOne, a miniature video camera which records 640 x 480-pixel frames at 28 frames/second to an SD card. The camera is powered from the controller, so that it can be switched on or off by radio control.

5. ROBOFOIL APPLICATIONS

The intention is that robofoil can be adapted quickly for a variety of applications, so it is essential that processing software is equally adaptable. To make this possible, we have been exploring ways of constructing image analysis applications purely by learning from examples. This is achieved using the machine learning technique known as *genetic programming* (GP) – see (Koza, 1990). This is closely related to the familiar genetic algorithm but while the latter optimises a set of numerical parameters, GP optimises what is essentially a program. Both techniques iterate towards a solution using operators that mimic evolution in the natural world.

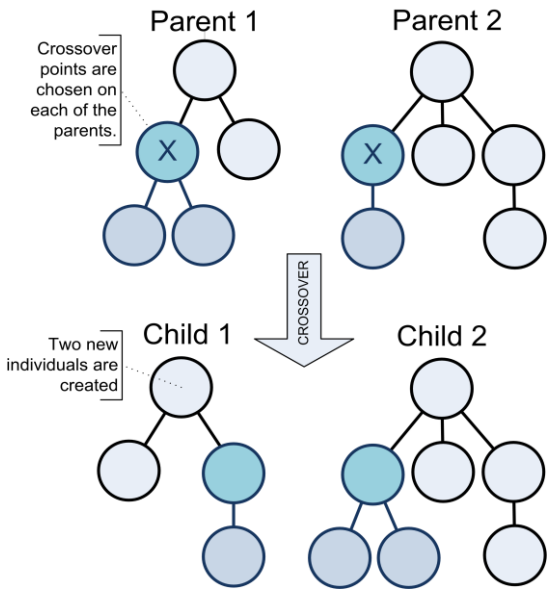

Figure 5. Crossover in GP. A point is selected as the crossover site in both parents and sub-trees are swapped to yield two distinct offspring.

Training data are gathered by means of a graphical interface in which the user identifies some pixels of each region of interest and associates a class label with each region. A similar but disjoint test set is also captured in the same way; this is used for assessing how well the programs that result from GP work on unseen data.

The learning process starts with a carefully-chosen set of image processing operators that effectively perform colour- and feature-based segmentation, and a similar set that calculates parameters of segments; further operators perform arithmetic, logical and other operations; a random population of programs is then generated. Each program is then executed on the training set, with its effectiveness measured by means of a *cost function*. Here, the cost function simply counts the number of correct results obtained. The most successful programs are then combined using processes that mimic sexual reproduction. Firstly, *crossover* exchanges randomly-chosen sub-trees of programs between individuals (Figure 5), while *mutation* replaces a randomly-chosen sub-tree with a randomly-generated one. Taken over the whole population, crossover tends to find minima in the cost function while mutation tends to jump out of local minima. The most successful programs are also allowed to persist into the next generation (known as *elitism*). When a new population has been constructed, the effectivenesses of the programs are measured via the cost function...and so on. Although this process involves a number of random selections in a number of places, it has been found to be both effective and robust.

GP has a reputation for being slow but there are some ways in which its speed can be improved. Firstly, strongly-typed GP (Montana, 1993) ensures that only operators that are compatible in terms out inputs and outputs are connected together. Secondly, and more importantly for image analysis, the overall image analysis problem is split into separate segmentation and classification stages. Finally, a novel training régime is used that saves individual programs that are able to classify one class of test data correctly in all the training set – see (Oechsle & Clark, under review) for details.

This approach has been applied to several problems in the remote sensing domain. An interesting one in the context of using UAVs is traffic monitoring: given an aerial image of traffic on a road, can one identify the locations of individual vehicles? Previous work has used GP to evolve a rotation-invariant object detector for recognising various classes of vehicle from infra-red imagery (Roberts and Howard, 1999). To evolve a vehicle detector with a degree of robustness, a set of images of vehicles on roads was captured from the Web, and some 600 cars were identified on them; half of them were used for training and the other half for testing. A vehicle segmenter was evolved, operating on grey-scale versions of the images only, and a classifier was evolved on the resulting labelled regions. Results on a typical unseen test image are shown in Figure 6; the sensitivity of the evolved system was 97.6% and the specificity was 91.8%.

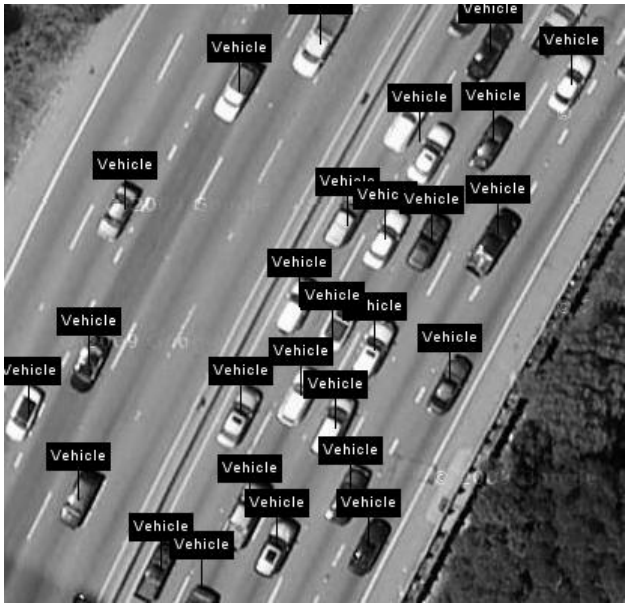

Figure 6. Result from the evolved vehicle detector on an unseen test image

The entire procedure – training and test data mark-up and evolution of the two stages – took about half a day and involved no custom-written software. This shows how well this learning-by-example approach is suited to the need for non-experts to develop automated image analysis capabilities in the field as the need arises.

6. CONCLUDING REMARKS

Our aim is to build a low-cost UAV that can be operated and maintained in remote regions yet provides useful mission capability. The existing design achieves this as long as robofoil is controlled from the ground. The system is also able to fly routes denoted by GPS waypoints autonomously, though the autopilot that makes that possible is considered to be too expensive. One of our major future aims is to replace this with a home-brewed system that provides equivalent functionality at a greatly reduced cost.

We also intend to investigate a somewhat more substantial image capture capability, based on a combination of an EeePC mini-laptop and webcams. We are working with Logitech Pro9000 webcams, which are able to capture either 640 x 480-pixel video or 1600 x 1200-pixel still frames, selectable via software.

The use of genetic programming to evolve automatic analysis capabilities purely through training complements the general aim of the robofoil work, to provide a hardware and software toolkit for remote sensing by UAV.

References

CAA, 2009. *Unmanned Aerial Vehicle Operations in UK Airspace – Guidance*, third edition (Civil Aviation Authority, London, UK).

R. De Nardi, O. Holland, J. C. Woods, and A. F. Clark, 2006. SwarmAV: A swarm of miniature aerial vehicles. In *Proceedings of the 21st Bristol International UAV Systems Conference.*

R. Dunford, K. Michael, M. Gagnage, H. Piégay and M.-L. Trémelo, 2009. Unmanned Aerial Vehicle Technology: Potential and Constraints for Riparian Vegetation Mapping. In: *Proceedings of the RSPSoc Annual Conference*, Leicester, UK, pp. 622-629.

A. P. Gay, T. P. Stewart, R. Angel, M. Easey, A. J. Eves, N. J. Thomas, D. A. Pearce and A. I. Kemp, 2009. Developing Unmanned Aerial Vehicles for Local and Flexible Environmental and Agricultural Monitoring. In: *Proceedings of the RSPSoc Annual Conference*, Leicester, UK, pp. 471-476.

J. R. Koza, 1992. Genetic Programming: On the Programming of Computers by means of Natural Selection. MIT Press.

D. G. Lowe, 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision **60**(2), 91-110.

I. F. Mondragon, P. Campoy, and J. F. Correal, 2007. Visual model feature tracking for UAV control. In: *IEEE International Symposium on Intelligent Signal Processing*, 1-6.

D. J. Montana, 1993. Strongly Typed Genetic Programming. *Evolutionary Computation* **3**, 199-230.

O. Oechsle and A. F. Clark, 2010. Towards Generic Computer Vision: Evolving Task-Specific Solutions From Generic Components Using Genetic Programming. Submitted to *IEEE Transactions on Pattern Analysis and Machine Intelligence.*

OpenPilot. A free software autopilot for small UAVs. http://www.openpilot.org/ (accessed 9 April 2010).

D. Howard and S. C. Roberts, 1999. Evolution of Vehicle Detectors for Infra-Red Linescan Imagery. *Lecture Notes in Computer Science* **1596**, 110-125. Springer-Verlag.

G. Zhou, V. Ambrosia, A. J. Gasiewski and G. Bland, 2009. Foreword to the Special Issue on Unmanned Airborne Vehicle (UAV) Sensing Systems for Earth Observations.
IEEE Transactions on Geoscience and Remote Sensing **47**(3), 687-689.