J Intell Robot Syst (2010) 57:171–191
DOI 10.1007/s10846-009-9378-y

A Cost Effective Tracking System for Small Unmanned Aerial Systems

Michail Kontitsis · Kimon Valavanis

Received: 1 February 2009 / Accepted: 1 August 2009 / Published online: 14 November 2009

Abstract In this work we address the problem of object tracking in a largely unknown dynamic environment under the additional constraint of real-time operation and limited computational power. The main design directives remain that of real time execution and low price, high availability components. It is in a sense an investigation for the minimum required hardware and algorithmic complexity to accomplish the desired tasks. We present a system that is based on simple techniques such as template matching adapted for use in a dynamically changing environment. After development, the system was evaluated as to its suitability in a traffic monitoring application where it demonstrated adequate performance.

Keywords Unmanned systems · Vision · Tracking · Uncalibrated camera · VTOL

1 Introduction

In recent years unmanned aircraft systems (UAS) have been successfully used in a wide variety of applications. Their value as surveillance platforms has been proven repeatedly in both military and civilian domains. As substitutes to human inhabited aircraft, they fulfill missions that are dull, dirty and dangerous [1]. Representative examples of successful use of UAS are in areas including battlefield assessment, reconnaissance, port security, wildlife protection, wildfire detection, search and rescue

M. Kontitsis (✉) · K. Valavanis (✉)
Department of Electrical and Computer Engineering,
School of Engineering and Computer Science,
University of Denver, Denver, CO, USA

M. Kontitsis
e-mail: mkontits@mail.usf.edu

K. Valavanis
e-mail: Kimon.Valavanis@du.edu

missions, border security and patrol, resource exploration and oil spill detection, to name just a few. The main common component among all those diverse applications is that they are practically variations of remote sensing and surveillance missions.

The reliance of almost every Unmanned Aerial Vehicle (UAV) application on the ability to sense, detect and track objects from a distance has motivated this paper, attempting to further investigate this issue. In particular, among the various types of UAS, small scale unmanned rotorcraft or Vertically Take-off and Landing (VTOL) vehicles have been chosen to serve as the sensor carrier platforms because of their operational flexibility. Having the advantage of being able to operate from almost anywhere, since they require little to none preexisting infrastructure, outweighs their deficit of speed and endurance when compared to their fixed wing counterparts. Their ability to hover and fly in relatively confined spaces makes them almost ideal for deployment at low altitude and in urban settings in which the majority of fixed wing platforms would be challenged to operate. Therefore, and although reported research findings are general enough, the focus of the paper is on designing and implementing an object tracking system for a small unmanned custom made VTOL vehicle.

To accomplish the aforementioned tasks autonomously, any UAS must be equipped with the appropriate sensors to collect data and have enough on-board processing power for data interpretation and decision making. It must also employ a collection of algorithms capable of dealing with a variety of tasks.

Cameras have been used as part of the UAS's sensor suite primarily as data collection equipment rather than navigational aids. Their function usually is to passively sense the environment for the presence of a specifically defined object of interest, record and transmit visual data back to a ground station for evaluation. As previously stated, an essential ability of an autonomous aerial vehicle is that of recognizing and tracking objects of interest, thus, keeping them within the field of view of the camera while recording their trajectory. This enhances the utility of the unmanned vehicle and facilitates the work of the ground control personnel. It allows the UAV to be used as a surveillance tool that expands the covered area without requiring constant attention. However, tracking arbitrary objects from an overflying moving platform in an uncontrolled environment can be extremely challenging given the variability of the parameters that influence the process. In an outdoors environment varying lighting conditions, unstructured clutter, motion, random occlusions and visual obstructions must be dealt with by the detection and tracking algorithms. A very important design directive for an autonomous UAV is the requirement of real-time operation. All the tasks must be completed as fast as possible. In the worst case, the computation time of the decision making components must not exceed the 33 ms barrier that is considered to denote real-time performance. An additional constraint is imposed on the algorithm by the carrying platform itself. Small aerial vehicles set a bound on the electrical power that can be carried along, which in turn limits the available processing power. With limited processing power at hand, the complexity of the algorithms becomes an important factor. Between algorithms that accomplish the same task, the one with low complexity is always desirable. In this case, it is crucial that the selected algorithm be able to run in real-time on less than state of the art equipment.

Another line of distinction exists between systems that are designed so that the processing takes place on-the-ground station and the ones that use an on-board computer. Obviously the former are not affected by any payload limitations therefore allowing for more powerful computers to be used.

In this paper we address the problem of object tracking in a largely unknown dynamic environment under the additional constraint of real-time operation and limited computational power.

2 Method of Solution

To address the tracking problem a simple template matching algorithm based on a similarity measure such as the sum of absolute differences, was implemented. The template is being continuously updated to maintain its relevancy throughout the time period that the object it describes is being tracked. The updated template at any time k is a linear combination of the best matching image patch and the template at time $k-1$. Finally the tracking system has been designed so that it can concurrently accept input from both a human operator and an automated source such as another program.

3 Contributions

The paper's contribution to the area of vision systems for unmanned aerial systems is the design and implementation of a cost effective system capable of performing object tracking in real-time that:

- Requires minimal information about the dynamic environment in which it operates;
- Uses a single uncalibrated, not stabilized camera;
- Tracks multiple objects without requiring a-priori knowledge of or using any assumptions about their trajectories;
- Does not require an IMU.

The result is a system that can be assembled by commercially available hardware and can be configured to perform surveillance without calibration of the camera or detailed knowledge of the operating environment. It becomes evident that the use of an uncalibrated, not stabilized camera makes the problem very challenging and to some extend limits the accuracy of obtained results. However, this is one major point addressed in this work: even with an uncalibrated, unstabilized camera, results are sufficient to complete assigned missions.

4 Paper Outline

This paper consists of five sections. The first section introduces the work and briefly describes the problem, the method of solution and the contributions. The second section provides a review of related work and some remarks on them. The third section is devoted to the detailed description of the proposed solution and the implemented system. The performance evaluation is presented in the fourth section along with a description of the actual scenarios where the system was deployed and the specific tasks that it carried out. Concluding remarks follow in the fifth section along with future research topics that can enhance the current implementation.

5 Related Work

Vision systems, techniques and algorithms suitable for UAVs range in complexity from simple color segmentation to statistical pattern recognition. This literature review considers a publication as being related to this work if the implemented system is specifically designed for use by UAVs. Furthermore, a work is considered directly comparable if the resulting vision system is physically placed on an unmanned VTOL/UAV and has been shown to function under real operating conditions in an outdoors environment.

Published related work and proposed machine vision architectures indicate the use of both “on-board” [2–9] and “on-the-ground” processing setups [10–19]. For on-board vision systems, due to the limited processing power provided by the on-board computer, derived algorithms have the additional constraint to run in real-time, requiring reduction of the computational load sometimes accomplished by processing selected image windows instead of the entire image. Table 1 summarizes

Table 1 Summary of system characteristics and functionality

Institution		Berkeley University	Georgia Tech	Univ. of South California	COMETS[a] [31]	WITAS[b] [8]	CNRS[c] [32]
Experimental setup	Dynamic observer	X	X	X	X	X	X
	Dynamic environment				X	X	
	Static/man-made environment	X		X			X
	Known landmarks	X	X	X			
	Natural landmarks				X		
	Calibrated cameras				X		
Capabilities	Depth mapping		X	X			
	Object identification	X	X	X	X		
	Object tracking		X	X	X		
Methods used	Optic flow			X		X	X
	Motion estimation	X			X	X	X
	IMU data						X
	Template matching	X		X	X	X	

[a]COMETS is a multi-national effort supported by the European Commission

[b]Wallenberg laboratory for research on Information Technology and Autonomous Systems (WITAS)

[c]Centre National de la Recherché Scientifique (CNRS) in France

functionality and capabilities of existing fully operational vision systems, including techniques employed by each one of them.

The problem of object tracking has been studied extensively in computer vision. Although several methods exist that can track objects in a controlled setting, special reference will be made to those of them that have been adapted for use in unmanned aerial systems since it is believed that they relate more closely to the problem at hand. One such example is illustrated in the work of Ludington et al. [20] that presents a method for target tracking using a technique based on particle filters. Each target is described as a four dimensional particle containing the image coordinates and the dimensions of the rectangle that surrounds it. The assumption for the system to operate is that the target moves smoothly between frames and that the frame rate remains sufficiently high.

The motion is modeled as Gaussian random walk and the measurement model relies on color and motion cues. Finally, a neural network is responsible for constructing a performance estimate according to which adaptations are made to the particle filter. Another example is the system described in [21, 22] where features such as rectangles are first extracted using the Douglas-Peucker algorithm [23] and then tracked using a Kalman filter.

Although not explicitly designed for an unmanned vehicle the system presented in [24] is addressing the problem of tracking moving objects in aerial video. It employs two different trackers; one for short and another for long term tracking. The short term tracking is accomplished by first registering successive frames using an affine transformation to correct for background motion and then extracting and matching Kanade–Lucas–Tomasi [25] features between successive image pairs.

The long term tracker relies on model matching to follow a specific pattern through the sequence. The Lucas–Kanade tracker is also utilized by Kanade et al. [26] in conjunction with a motion prediction scheme that relies on Kalman filtering, an image pyramid and two dimensional affine motion models to deal with large motion in image sequences taken from a micro-unmanned aerial vehicle.

Motion tracking using low cost off the shelf components is also investigated in [27] where a fixed wing UAV is relaying data back to a ground station where the processing takes place. The authors use a proprietary vision system which according to their accounts "lost target track on a regular basis". Tracking salient points is demonstrated in [28] using SIFT features and the RANSAC algorithm where the authors are reporting correct projection in 55.4% to 82.5% of the frames while spending 0.63 to 1.93 s per frame.

Another area where tracking a ground target is important is that of autonomous landing. As shown in [29] the helipad is usually marked by an "H" which is extracted from the images using fixed threshold segmentation and tracked during the landing maneuver by means of second and third order moments of inertia.

6 Challenges of Implementing Vision for a VTOL

Helicopters are attractive as unmanned vehicles due to their ability to take off from almost anywhere without the requirement of a runway. Furthermore, their ability to hover makes them ideal for surveillance since they can keep the onboard sensors pointed towards the same area without having to execute elaborate maneuvers. The price to pay for that flexibility is their lower speed, limited endurance and inherent instability when compared to fixed wing aircraft. Small unmanned helicopters are even more unstable and susceptible to even modest changes in environmental conditions. This instability affects the images that any onboard camera would acquire and requires either the use of stabilization hardware which adds weight or stabilization software, which adds in turn complexity and demands more processing power.

The unmanned aerial vehicles that were used for the purposes of this work are designed to operate outdoors. Such an environment is notoriously difficult for computer vision primarily due to variations in lighting. Furthermore, there is usually limited availability of a-priori knowledge of the environment and certainly no three dimensional map which leaves little room for helpful assumptions to be made.

The low cost requirement and the low payload allow only for a single camera to be carried on-board. A second camera could have been utilized to allow for a stereoscopic system and provide additional data for verification purposes leading to a more robust detection. Its absence can be viewed as an additional design constraint for the vision algorithms.

7 Tracking System

The system is required to be able to track multiple objects through time. It is designed in a way that allows the user to designate the objects to be tracked during runtime. The main challenges include trajectory prediction as well as occlusion handling and motion segmentation. The demand for low computational cost, real-time execution, and a minimal object description model still applies.

Although the helicopter is perceived to be stationary when hovering and attempting to track ground bound objects, this is rarely the case. Given the relatively small size of both the Raptor 90 and the Maxxi Joker, even slight variations in the wind's direction or speed can result in an unstable flight profoundly influencing the quality of the acquired images. This translates to relatively high disparities between corresponding objects in subsequent image frames. Furthermore, it makes tracking objects close to the boundaries of the image almost impossible because they may appear and disappear erratically due to the relative motion of the camera with respect to them.

Occlusions present a significant challenge when attempting to track a specific object in a dynamically changing environment. Objects are expected to move almost randomly and therefore occlude each other. The background environment although static can contribute to this problem whenever it includes obstacles comparable in size with the objects of interest. Tree lines for example are such typical obstructions. Also, since the camera is mounted on a moving platform it is possible for any object to become occluded even by remaining stationary. As one might expect, occlusions are greatly reduced in frequency when the optical axis of the camera is perpendicular to the terrain. However hovering directly above the area of interest may not always be feasible or desirable due to safety concerns.

Motion or background segmentation is another challenge due to the nature of the environment that the unmanned vehicle operates in. Typical background extraction techniques such as frame differencing or Kalman filtering do not cope well with rapidly varying scenes. In particular, frame differencing degenerates to a crude edge detector when applied to a sequence of images acquired by a moving camera. On the other hand, motion estimation algorithms like the ones used in the MPEG standard that were also considered, found to be highly demanding in terms of processing power. However with dedicated hardware that accelerates MPEG encoding this could be a viable choice for motion estimation.

7.1 Tracking System Module Overview

Having made a review of the typical problems related to object tracking within the context of small unmanned VTOLs, we now describe the operation of our system along with the modules that constitute it. At first the object to be tracked has to be specified. This information can either come from an object detection system that is automatically searching for a pre-specified object or from a user who activates the tracking system by manually selecting an object in the image. The target selection module then creates a description of the selected object and forwards it to the matching module. The latter will attempt to find a corresponding template in the subsequent image. Once such a match is found the original template that describes the object is updated with information from the most currently acquired frame. Finally the Pan-Tilt controller signals the Pan-Tilt mechanism to move accordingly so that the tracked object is at the center of the image. Briefly stated the tracking system is comprised of:

- The target selection module.
- The matching module.
- Template update module.
- Pan-Tilt controller.

A block diagram showing the interconnections between the aforementioned modules can be seen in Fig. 1. The operation of each of the modules at any point in time is described in the following paragraphs and shown in Fig. 2.

7.2 Target Selection Module

This component is responsible for receiving the user's input and creating a description for the object to be tracked. In both cases the input to this module is a series of

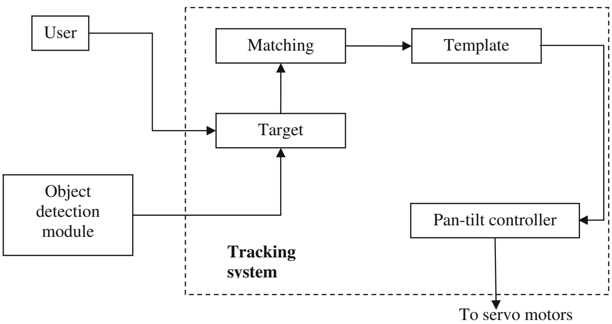

Fig. 1 A block diagram of the tracking system

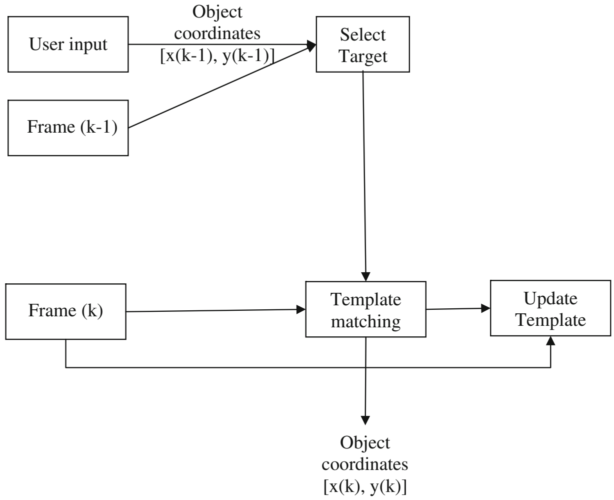

Fig. 2 Operation of the tracking system at a given time *k*

coordinates (x, y) representing the object's position in the image's column and row coordinate system as seen in Fig. 3. The design choice has been made to employ a $N_t \times N_t$ area taken around the image point (x, y) as a template for matching into subsequent image frames.

7.3 Matching

This module attempts to find a match for the template within the current frame. To reduce the search space for this match the search is limited within a ($N_{search} \times N_{search}$) area around the latest previously known position. Assuming continuity of motion for the object under tracking, it is reasonable to expect that it will appear in the next frame relatively close to its current position. The similarity measure for the match

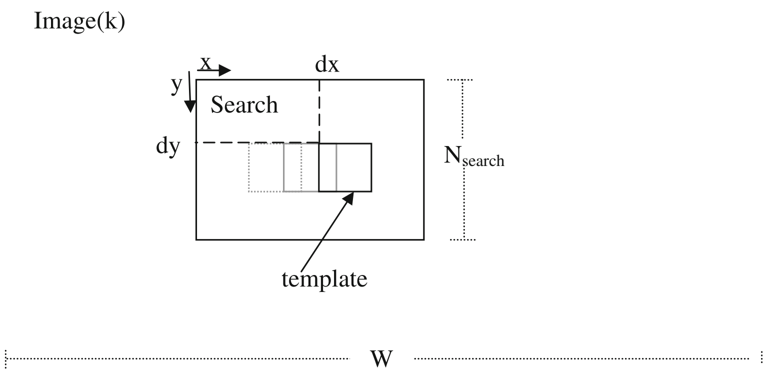

Fig. 3 The template matching process

is the sum of absolute differences between the template and all the $N_t \times N_t$ square sub-images within the search space:

$$SAD\left(d_x, d_y\right) = \sum \left|template\,(x, y) - image_k\left(x + d_x, y + d_y\right)\right| \tag{1}$$

The row and column position that minimize the similarity measure become the output of this module.

$$\underset{dx,dy}{\arg\min}\left(SAD\left(d_x, d_y\right)\right) \tag{2}$$

Obviously a trade off exists when selecting the size of the search space. A large value for N_{search}will allow for a larger disparity between the positions of the object in subsequent frames. The penalty for a larger search space is obviously the extra computational cost which increases with the square of N_{search}. On the other hand, decreasing the search space may save some computing time but it entails the possibility of not finding a proper match just because the object moved more than $N_{search}/2$ pixels in any direction. A good compromise was achieved by making a selection based on the maximum expected apparent motion *Max_disparity*.

For our applications we selected: $N_{search} = 2 \times Max\_disparity = W/10$, where W is the width of the captured image. Our implied assumption is that the apparent motion of the observed object will not be exhibiting inter-frame displacements of more than $W/20$ pixels. Figure 3 shows the relation between the sliding template, the image and the search space.

7.4 Template Update Module

The typical weakness of a tracking system based on template matching is the fact that with time the template may become irrelevant. Objects are moving and their pose with respect to the camera is almost constantly changing. As a result, the projected two dimensional image of any given object differs significantly within the time span of a few seconds making any attempt for a match with the original template almost impossible.

To mitigate this effect the template is updated at every cycle of the algorithm's execution. Every new captured image, within which a match was found, contributes to the template by introducing information to it thereby forming a new one. The new template is a linear combination of the existing one and the neighborhood of the best match. In most cases the linear combination of the current template and the best matching patch is sufficient to maintain the relevancy of the template without incurring a significant processing power penalty.

If $Template(k)$ is the template at time k and $Match(k)$ the $N_t \times N_t$ neighborhood around the coordinates of the best match then:

$$Template\,(k+1) = a\ Match\,(k) + (1-a)\ \ Template\,(k)\,,\ \ where\ a \in [0, 1]\,.$$

The design decision to be made when calculating this new template is about the amount of new information that will be incorporated versus the amount that will be retained from the "old" template. Apparently there is a trade-off. If one chooses to retain more of the older template the result will be a slower changing template unable to accommodate pose variations that happen within the time span of a few frames. This, however, will make the template impervious to short duration random illumination variations as well as to the occasional miss-match. On the other hand if the choice is made to aggressively update the template with new information then it has a better chance of remaining relevant and being able to cope even with objects whose pose and appearance vary rapidly. The caveat in this case is that the template becomes susceptible to noise and to the fact that even a single mis-match by the matching module can easily throw-off the tracking system by forcing it to follow the new miss matched object. The balance between the new and old information is controlled by the constant a. Lower values of a place more weight on the old template rather than the newly acquired image and higher values of course have the opposite effect. After some experimentation the value $a = 0.025$ was chosen as the one that yielded the best compromise between robustness and adaptation.

7.4.1 Evaluating the Template Update Module

To further investigate the way the values of a influence the output of the tracking system, a series of experiments were conducted. Each time, the system was instructed to track the same object through the sequence while the value of a remained constant. This was repeated for values of a ranging from 0.005 to 0.995 with a step of 0.01. The sequence selected for that purpose is one that contains a car executing a u-turn

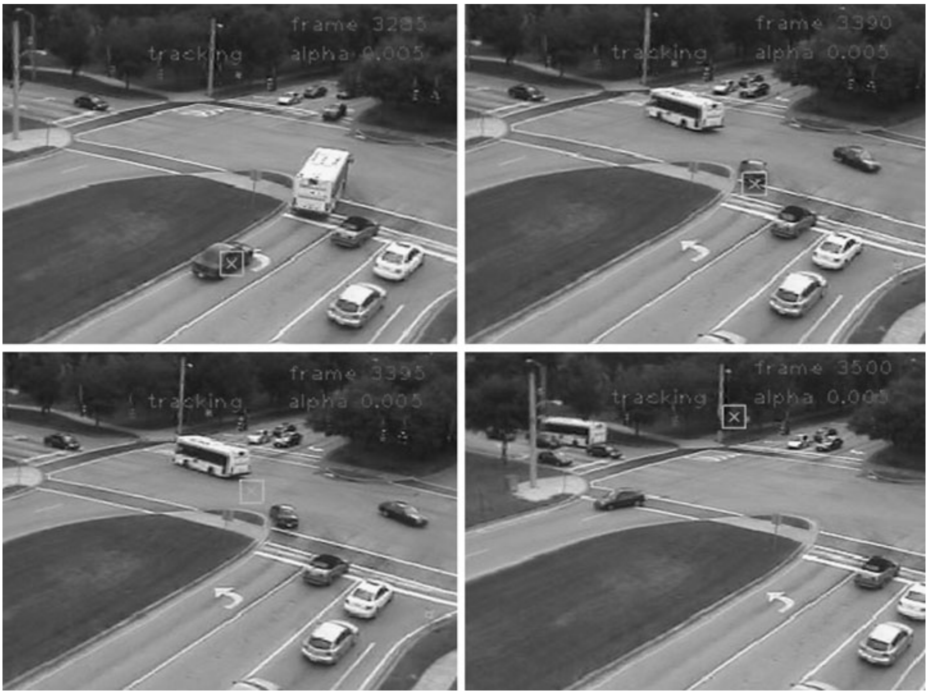

Fig. 4 Tracking results for a = 0.005

maneuver. The choice was made because that particular video excerpt contains an object that changes its pose relatively fast allowing the opportunity to validate the effectiveness of the way the template is updated to accommodate the changing appearance of the target. However, the template must also retain some of the past information to ensure the identity of the target. During the experiment, it was verified that for low values such as 0.005 the template does not adapt fast enough to maintain the track. Figure 4 shows exactly that. Conversely, values above 0.155 forced out of the template enough past information so that the tracking system abandoned the initial target as shown in Fig. 5.

Finally, the value of *a* for which the system exhibited the desired behavior was 0.025. As shown in Fig. 6 the vehicle is consistently tracked through the sequence. Having established this trajectory as the ground truth, the root mean square (RMS) error was calculated for the ones produced by each of the 100 different values of *a*. Isolating some characteristic values of *a* and plotting the error for them yields Fig. 7. One can notice the sharp increases that correspond to the time that the tracking failed.

7.5 Pan–Tilt Controller

It is usually desirable, if not required, that the tracked object remains in the camera's field of view (FOV). This task is accomplished by the Pan–Tilt controller, which as the name implies, sends control signals to the servos that adjust the pan and tilt angles

Fig. 5 Tracking results for a = 0.155

Fig. 6 Tracking results for a = 0.025

of the camera. The goal is to keep the tracked object approximately at the center of the captured images. For that reason an error vector is calculated between the center of the image and the image point where the object is located. The vertical and

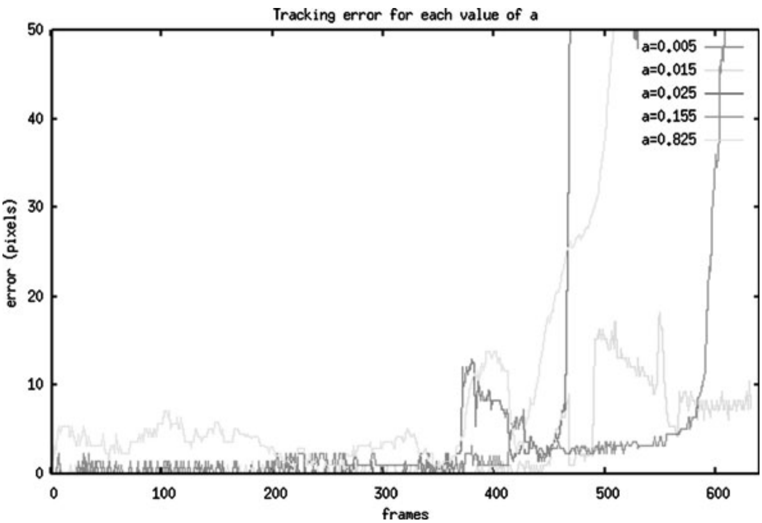

Fig. 7 Plot showing the RMS error for some characteristic values of *a*

horizontal components of this error vector are then used to adjust the pan and tilt so that the error is minimized. The control rules for the Pan–Tilt are:

If $error_x < -N$ then pan left by 5°
If $error_x > N$ then pan right by 5°
If $error_y < -N$ then tilt up by 5°
If $error_y > N$ then tilt down by 5°

To avoid oscillation and constant corrections the object is kept within a $N \times N$ window centered around the center of the image.

8 Traffic Monitoring

A very interesting application for an unmanned VTOL is that of traffic monitoring. The VTOL is a surprisingly suitable platform since it can hover over a particular traffic node for varying periods of time depending on configuration. A medium sized autonomous helicopter such as the Bergen Observer can hover for a period of 45 min and up to more than an hour and a half if equipped with extra fuel tanks. However the electric powered Maxxi Joker 2 can only provide a 12 to 15 min hover, although this is expected to increase as new, higher density battery technology becomes available. Furthermore, a traffic monitoring system based on autonomous or semi-autonomous VTOLs can be deployed very rapidly in areas that provide little or no infrastructure. Once deployed, it can provide real-time data to operators on the ground or to traffic simulation models which can then provide more accurate predictions for traffic parameters based on more current observations. The operator of the system can also direct the system to track a certain vehicle by manually selecting it with a pointing device (e.g. a mouse). Additionally, such a system can serve as an ad-hoc replacement for already existing traffic monitoring infrastructure in case the latter has failed or has been accidentally destroyed. In emergencies such as a hurricane evacuation or a large scale automotive accident an autonomous VTOL deployed from a first responder's vehicle can provide helpful information.

Briefly stated, the autonomous VTOL provides the capability of a traffic monitoring system without the requirement of extensive infrastructure. It has two modes of operation, one automatically extracting traffic data and another tracking manually selected vehicles.

9 Description of the System

As stated above one of the objectives of the system is to extract meaningful real-time data from the video stream captured by the onboard camera on the VTOL. Such data include the total number of vehicles that pass through a given part of the road network, the number of vehicles that follow a certain path and the overall traffic flow. This task can be separated into three distinct steps. Initially the image areas that correspond to vehicles have to be differentiated from the environment. Secondly, the extracted vehicles must be consistently tracked as they traverse the portion of the road network that is being examined. Lastly, the result of the tracking procedure is converted to meaningful traffic measures. A block diagram showing the succession

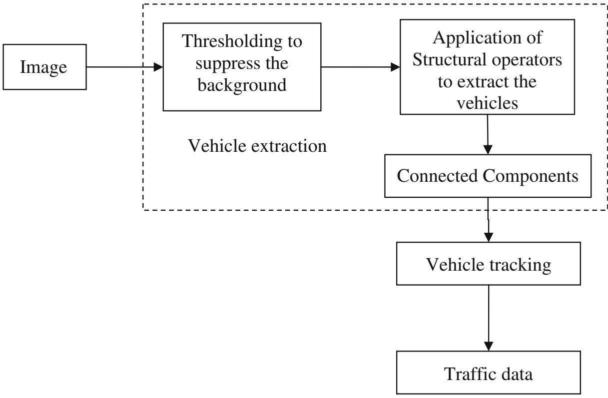

Fig. 8 A block diagram representation of the data extraction mode of the traffic monitoring system

of these steps is shown in Fig. 8. The following paragraphs provide a more detailed description of the whole process.

The first step towards extracting the desired traffic data involves identifying the vehicles. In keeping with the goals of this work, the vehicle extraction process must be simple, computationally cheap and requiring minimal a priori knowledge. The selected method takes advantage of the fact that the areas of the image corresponding to a paved road usually have extremely low saturation values. A simple sufficiently low threshold is then applied to suppress the part of the background that is the road. Following that, a pair of erode/dilate morphological operators are used to further distinguish the blobs that correspond to vehicles. A size filter is employed to avoid some residual areas of the background being categorized as vehicles. In particular, for any formation of pixels to be accepted as a vehicle it has to fit within a bounding rectangle no smaller than $W/10 \times H/10$ and no larger than $W/3 \times H/3$, where W and H are the width and height of the image respectively. Its effectiveness is based on the assumption supported by the observation that areas smaller than $W/10 \times H/10$ usually correspond to noise whereas areas with dimensions larger than $W/3 \times H/3$ are usually representative of various background formations. The application of such a filter enforces a measure of scale on the problem albeit a reasonable one. Figures 9,

Fig. 9 The RGB input image in the *left* is converted to HSI. The saturation component is shown in the *right*

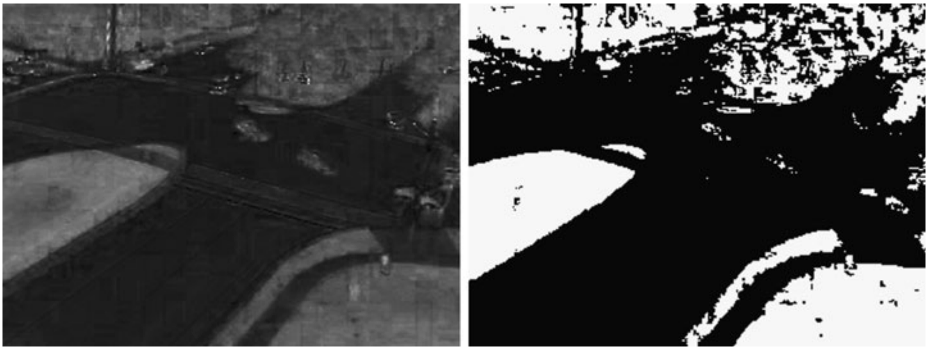

Fig. 10 The application of a threshold on the saturation component (*left*) eliminates most of the pixels belonging to the road (*right*)

10, 11 and 12 depict the image processing steps leading from the input image to the extracted vehicles.

In succession, a set $track_{k-1}$ is constructed containing the center of gravity of all the regions that pass the size filter at a given time $k-1$. This set containing pairs of (i, j) image coordinates is presented as input to the tracking module described

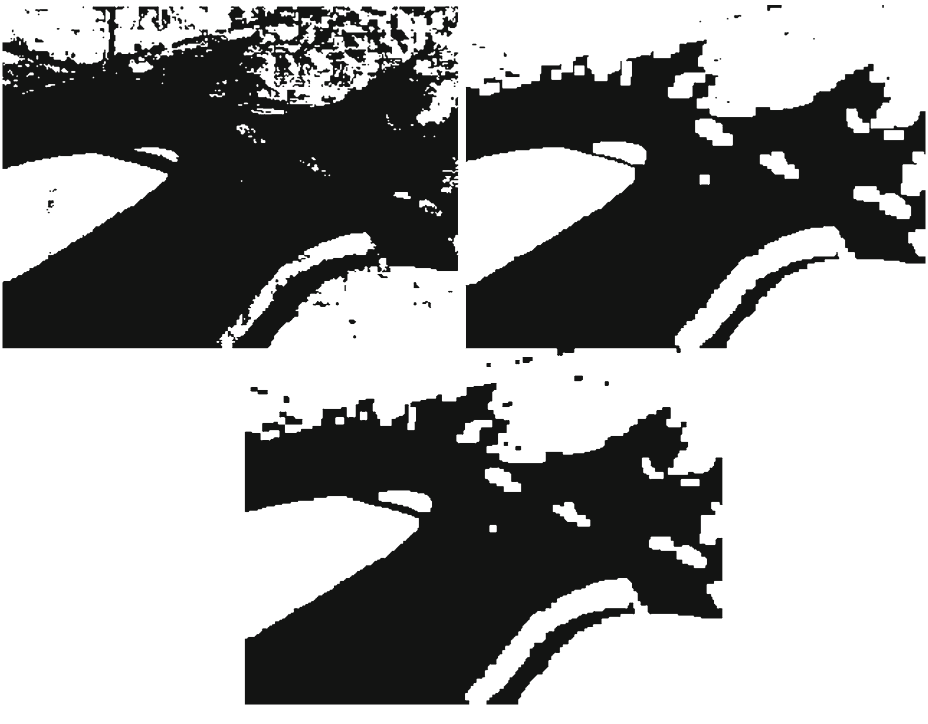

Fig. 11 Applying morphological operators to the binary image (*upper left*). Dilation (*upper right*) then erosion (*lower center*)

Fig. 12 Extraction of regions using connected components and a size filter. Notice the rejection of very large and very small blobs

in Section 3. The output of the tracking module forms a set $track_k$ that holds the respective matching coordinates at the current frame k. It is worth mentioning that time in this case has been discretized to coincide with the acquisition rate of the camera. So frame k and time k are interchangeable.

Having completed the vehicle extraction and tracking the remaining task is to calculate the traffic parameters of choice. The number of vehicles currently present on the road is simply equal to number of objects represent in the tracking set. In other words the cardinality of $track_k$ gives the number of current vehicles. The amount of vehicles over any given period of time can be found by integrating the function of vehicles over time.

Plotting the function of vehicles over time, results in the graph shown in Fig. 13. The red line represents the number of vehicles that the system estimates are currently

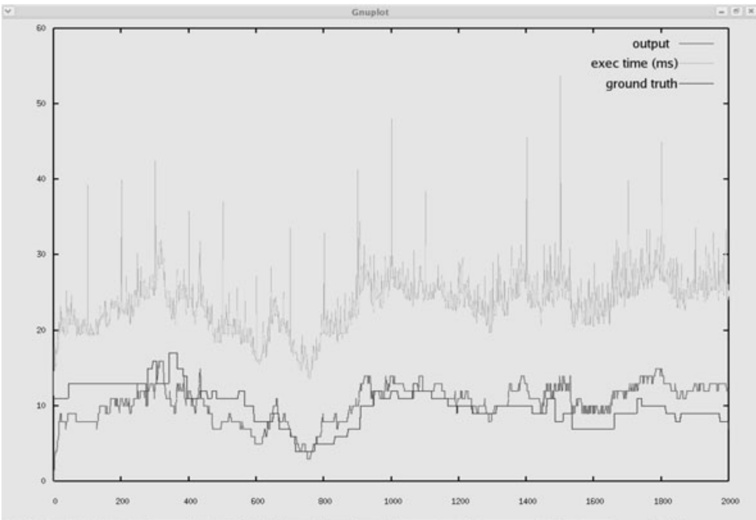

Fig. 13 The output of the traffic load estimator compared to the ground truth

in view, the blue line is the actual number of vehicles present and the green line shows the execution time in milliseconds. On average the system estimated the current traffic with an accuracy of 81%. For comparison, traffic monitoring systems that are utilizing fixed cameras [30] have reported higher rates of 95% to 97%. They benefit from the more consistent background and motion extraction since the sensors remain static and all apparent motion in the image can only be attributed to vehicles. Note that the execution time remains well below 33 ms which signifies real-time operation since it coincides with the capture period of the camera. The occasional spikes correspond to write cycles during which the operating system logged the output of the vision program to the disk. They are not due to executing the core vision code

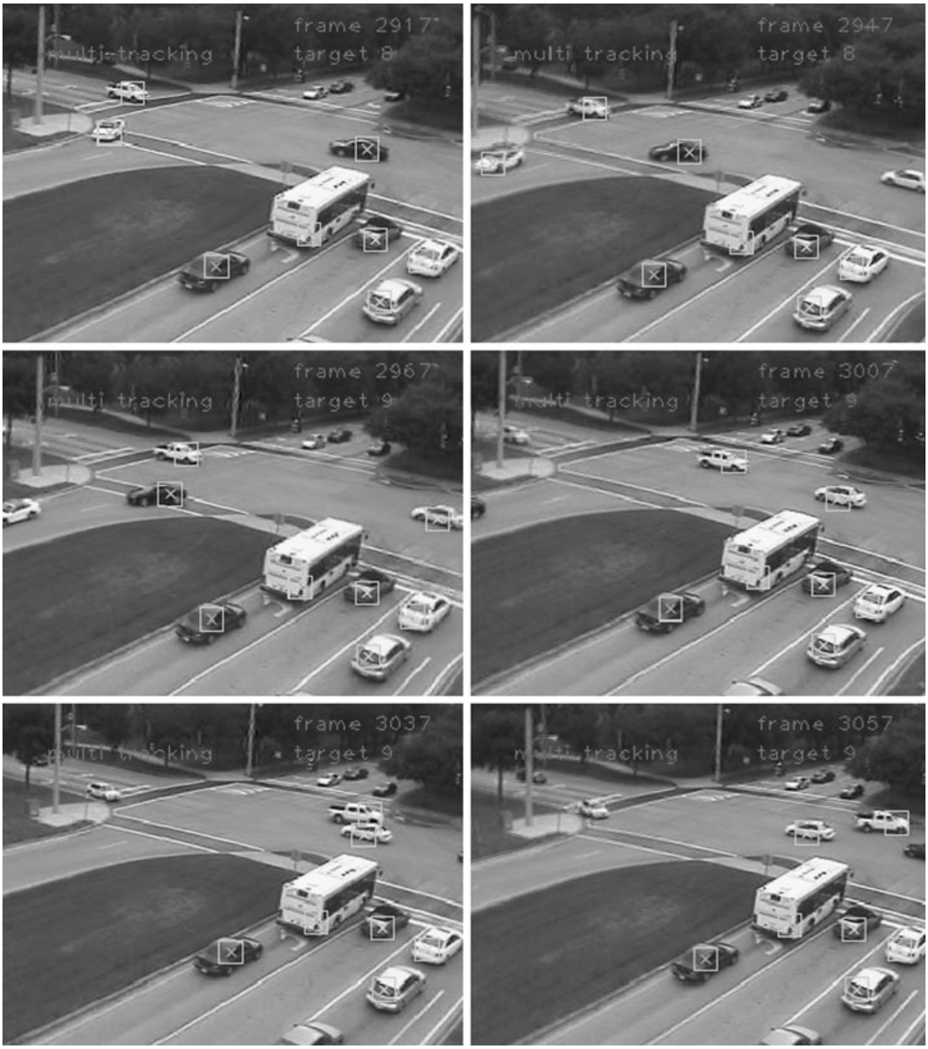

Fig. 14 Tracking is maintained despite the unpredictable motion of the VTOL and the parallax inducing motion of the vehicles

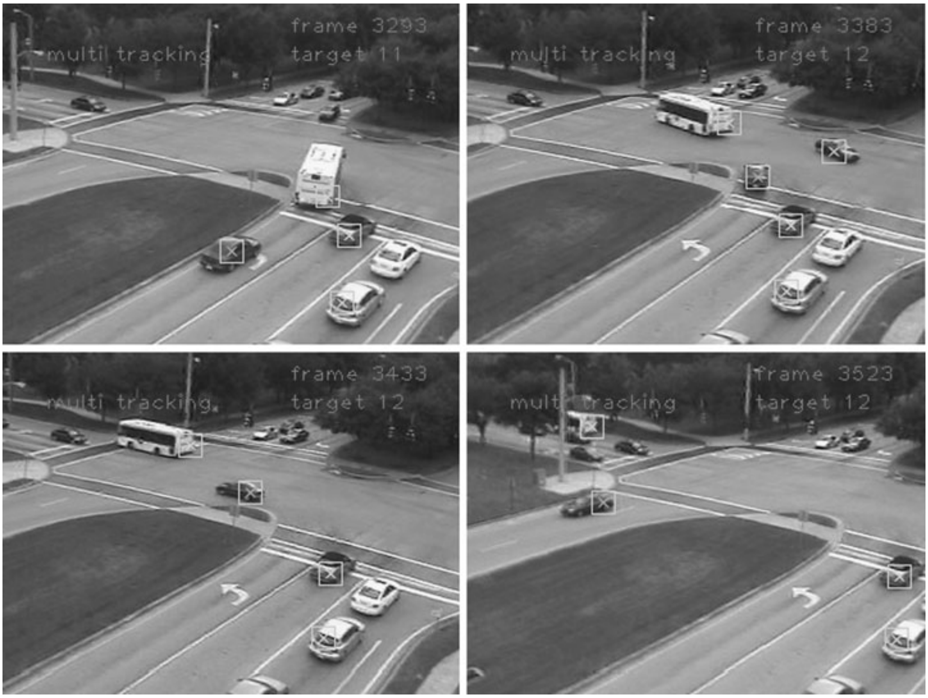

Fig. 15 Tracking of a vehicle making a u-turn

and are not present during normal operation. The computer used in this case was a Pentium 4 at 3 GHz.

The system can also function in another mode of operation that relies on user input for target designation. It allows a certain number of manually selected vehicles to be tracked. The operator has only to point and click on the vehicles to be tracked. The image coordinates are gathered and presented as input to the tracking system described in Section 3. Targets are abandoned by the system as they exit the field of view in order to make the resources of memory and processing power available for reallocation to tracking newly acquired targets. Figure 14 shows that the tracking can be maintained despite the unpredictable motion of the VTOL and the parallax inducing motion of the vehicles. Further resilience to parallax is demonstrated in Fig. 15 where vehicles are shown to be tracked while executing turns and u-turns that change their pose with respect to the camera.

10 Conclusion

This paper presented a system for object tracking to be used by unmanned aerial systems. It is based on template matching using the sum of absolute differences as a similarity measure. A template update occurs in every loop of the algorithm by incorporating new information so that the former remains relevant allowing the track to continue despite changes in the object's appearance. Furthermore it allows for manual entry of targets by the human operator as well as receiving input from other image processing modules. The following paragraphs further discuss the results and contributions of this paper as well as provide some final remarks.

11 Discussion

The tracking system developed in this work was able to correctly track on average 81% of the visible vehicles when assigned to the task of traffic monitoring. It was shown capable of coping with significant disturbance, resulting from the nature of the carrying platform and the unstabilized camera, as well as with occasional brief occlusion and change of pose of the tracked object. However, there are specific situations that are beyond the ability of the system to compensate for. A severe disturbance that results in an apparent motion of more than W/20 between successive frames, where W is the width of the frame, will also result in a lost track. For an image acquisition rate of 30 fps, this apparent motion amounts to 1.5 W pixels/s which is arguably a high enough limit. Reversely, assuming that the camera is perfectly stable, only objects able of covering the largest visible dimension in two thirds of a second will avoid being tracked. Another case where the track can be lost is when an occlusion occurs that covers more than 50% of the template corresponding to the object being tracked. In such a circumstance, within a few repetitions of the template update process the template will lose enough information to make finding the proper match problematic.

This work contributes to the area of vision systems for unmanned aerial systems by proposing a monocular, uncalibrated, not gyro-stabilized design capable of identifying and tracking multiple objects in real-time, without strict assumptions about their trajectories, by relying on minimal information about the environment while forgoing the need for motion compensation usually requiring an inertial measurement unit (IMU). By selecting a monocular, non stabilized system the design avoids the extra cost and weight of a binocular configuration along with that of a gyro-stabilized turret. The fact that the camera does not require calibration for the system to operate, further simplifies the assembly and setup process.

The ability to track multiple objects at a low computational cost allows the design to run in real-time on inexpensive, less powerful, power efficient computing platforms. This is especially critical for relatively small unmanned with limited payload capabilities for which carrying a large energy reserve to support a powerful computer is impossible. Although the prevailing trends in computer design and battery chemistry promise to alleviate this problem by providing more efficient, powerful hardware and higher energy density batteries respectively, the desire for the minimalist's approach in the design of a vision system will most likely continue to be relevant. This is especially true for the class of unmanned aerial systems that are even smaller than the Maxxi Joker 2 platform that was used in this work.

By keeping the requirement of knowledge about the environment to a minimum, the design is not explicitly required to perform elaborate background extraction thus further reducing the computational burden. No a priori known landmarks are used to facilitate detection and tracking and minimal assumptions, such as the one used in the traffic monitoring scenario about the tarmac being gray, are made. This allows for a fairly versatile system capable of performing in different environments with only a few, if any, modifications.

Future improvements should include an adaptive search space for the tracking system based on probabilistic metrics. Also, the availability of relatively inexpensive, lightweight cameras with electronically controlled optics offers the opportunity to further explore the problem from the perspective of active vision. Furthermore, varying optics can be employed by an attention focusing mechanism to extract more information in cases where disambiguation between objects is required. This zooming capability can also be of significant help to the human operator who can employ it to further investigate an object of interest without interfering with the flight of the VTOL.

In summary, this work produced a design suitable for tracking multiple objects in real-time. It demonstrated that it is possible to endow relatively small, inexpensive VTOLs with capabilities usually reserved for more expensive, higher end unmanned systems.

Acknowledgement This work has been partially supported by NSF Grant, IIP-0856311 (DU grant number 36563).

References

1. Unmanned systems roadmap: 2007–2032. Office of the Secretary of Defense (2007)
2. Vision hardware. University of Southern California. http://www-robotics.usc.edu/∼avatar/vision_hw.htm (2004). Accessed 27 Jan 2005
3. Mejias, L., Saripalli, S., Sukhatme, G., Cervera, P.: Detection and tracking of external features in an urban environment using an autonomous helicopter. In: IEEE International Conference on Robotics and Automation (2005)
4. Hrabar, S., Sukhatme, G.S.: A comparison of two camera configurations for optic-flow based navigation of a UAV through urban canyons. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2673–2680 (2004)
5. Sattigeri, R., Calise, A.J.: An adaptive approach to vision based information control. In: Guidance, Navigation and Control Conference, no. AIAA-2003–5727, Austin, TX (2003)
6. Enhanced Vision System (EVS) overview, CMC electronics. www.cmcelectronics.ca/En/Prodserv/Commav/commav_evs_overview_en.html (2001). Accessed 22 Jan 2005
7. Groven, J., Holk, E., Humbert, C., Krall, J., Schue, D.: Rose-Hulman institute of technology autonomous helicopter for the 2004 international aerial robotics competition (2004)
8. Nordberg, K., Doherty, P., Farneback, G., Forssen, P.-E., Granlund, G., Moe, A., Wiklund, J.: Vision for a UAV helicopter. In: 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems—IROS 2002. Proc. Workshop WS6 Aerial Robotics, pp. 29–34, Lausanne (2002)
9. Anderson, J.D., Lee, D.-J., Edwards, B., Archibald, J., Greco, C.R.: Real-time feature tracking on an embedded vision sensor for small vision-guided unmanned vehicles, pp. 55–60. http://ieeexplore.ieee.org/iel5/4269829/4269830/04269882.pdf (2007). Accessed 20 July 2007
10. Sastry, V.: Vision based detection of autonomous vehicles for pursuit evasion games. In: 15th Triennial World Congress, Barcelona, Spain (2002)
11. Hrabar, S., Sukhatme, G.S.: Omnidirectional vision for an autonomous helicopter. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 3602–3609 (2003)
12. Saripalli, S., Montgomery, J.F., Sukhatme, G.S.: Vision-based autonomous landing of an unmanned aerial vehicle. In: Proceedings of IEEE International Conference on Robotics and Automation, pp. 2799–2804 (2002)
13. Woodley, B.R., Jones, H.L., LeMaster, E.A., Frew, E.W., Rock, S.M.: Carrier phase GPS and computer vision for control of an autonomous helicopter. In: ION GPS-96, Kansas City, Missouri (1996)
14. Debitetto, J.: Modeling and simulation for small autonomous helicopter development. In: AIAA-1997–3511 Modeling and Simulation Technologies Conference, New Orleans, LA (1997)

15. Woodley, B., Jones, H., Frew, E., LeMaster, E., Rock, S.: A contestant in the 1997 international aerial robotics competition. http://sun-valley.stanford.edu/papers/WoodleyJFLR:97.pdf (1997)
16. Holifield, N., Lallinger, J., Underwood, G.: International aerial robotics competition 2004. University of Texas at Austin, IEEE Robot Team Aerial Robotics Project 2004 http://iarc1.ece.utexas.edu/~lynca/final_documentation/utiarc 2004.pdf (2004)
17. Burleson, W., Salhany, W., Hudak, J.: Southern Polytechnic State University autonomous remote reconnaissance system. http://a-robotics.spsu.edu/SPSU_paper2005.pdf (2005)
18. Musial, M., Brandenburg, U.W., Hommel, G.: MARVIN—technische universität Berlin's flying robot for the IARC millennial event. In: Proc. Symposium of the Association for Unmanned Vehicle Systems 2000, Orlando, Florida, USA (2000)
19. RMAX Type I/ Type II G, Yamaha. http://www.yamahamotor.co.jp/global/business/sky/lineup/rmax/index.html (2004). Accessed 27 Jan 2005
20. Ludington, B., Reinmann, J., Vachtsevanos, G.: Target tracking and adversarial reasoning for unmanned aerial vehicles. In: Aerospace Conference, 2007 IEEE volume, Issue, pp. 1–17, 3–10 March 2007
21. Mejias, L., Saripalli, S., Campoy, P., Sukhatme, G.: Visual servoing of an autonomous helicopter in urban areas using feature tracking. J. Field Robot. **23**(3/4), 185–199, Wiley Periodicals (2006)
22. Mejias, L., Correa, J.F., Mondragon, I., Campoy, P.: COLIBRI: a vision-guided UAV for surveillance and visual inspection. In: Proceedings IEEE International Conference on Robotics and Automation, pp. 2760–2761, Rome, Italy (2007)
23. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of number of points required to represent a digitized line or its caricature. Can. Cartogr. **10**(2), 112–122 (1973)
24. Bell, W., Felzenszwalb, P., Huttenlocher, D.: Detection and long term tracking of moving objects in aerial video (1999)
25. Shi, J., Tomasi, C.: Good features to track. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 593–600 (1994)
26. Kanade, T., Amidi, O., Ke, Q.: Real-time and 3D vision for autonomous small and micro air vehicles. In: 43rd IEEE Conference on Decision and Control (CDC 2004) (2004)
27. Dobrokhodov, V.N., Kaminer, I.I., Jones, K.D., Ghabcheloo, R.: Vision-based tracking and motion estimation for moving targets using small UAVs. In: Proceedings of the 2006 American Control Conference, pp. 1428–1433, Minneapolis, Minnesota, USA, 14–16 June 2006
28. Mondragon, I.F., Campoy, P., Correa, J.F., Mejias, L.: Visual model feature tracking for UAV control. In: IEEE International Symposium on Intelligent Signal Processing, pp. 1–6 (2007)
29. Saripalli, S., Montgomery, J.F., Sukhatme, G.S.: Visually guided landing of an unmanned aerial vehicle. IEEE Trans. Robot. Autom. **19**(3), 371–380 (2003)
30. Wan, C.L., Dickinson, K.W.: Computer vision and neural networks for traffic monitoring. In: Road Traffic Monitoring 1992 (IEE Conf. Pub. 355), p. 143 (1992)
31. Ollero, A., Ferruz, J., Caballero, F., Hurtado, S., Merino, L.: Motion compensation and object detection for autonomous helicopter visual navigation in the COMETS system. In: IEEE International Conference on Robotics and Automation, New Orleans, LA (USA) (2004)
32. Ruffier, F., Franceschini, N.: Visually guided micro-aerial vehicle: automatic take off, terrain following, landing and wind reaction. In: Proceedings. ICRA '04. 2004 IEEE International Conference on Robotics and Automation, vol. 3, pp. 2339–2346, 26 Apr–1 May 2004