

Towards vision-based safe landing for an autonomous helicopter

Pedro J. Garcia-Pardo ^{a,*}, Gaurav S. Sukhatme ^{b,**} and
James F. Montgomery ^{b,**}

^a*DISAM, Universidad Politecnica Madrid, Spain*

^b*Robotic Embedded Systems Lab, Robotics Research Labs, University of Southern California, Los Angeles, USA*

Abstract

Autonomous landing is a challenging problem for aerial robots. An autonomous landing manoeuvre depends largely on two capabilities: the decision of where to land and the generation of control signals to guide the vehicle to a safe landing. We focus on the first capability here by presenting a strategy and an underlying fast algorithm as the computer vision basis to make a safe landing decision. The experimental results obtained from real test flights on a helicopter testbed demonstrate the robustness of the approach under widely different light, altitude and background texture conditions, as well as its feasibility for limited-performance embedded computers.

Key words: autonomous helicopter, vision-based landing, safe landing, contrast measures, texture scale

1 Introduction

Autonomous aerial vehicles bring together a large set of requirements that introduce many new research challenges to mobile robotics. In this paper, we focus on a key capability for aerial robots - the ability to land autonomously.

Helicopters are interesting aerial vehicles for research purposes. Due to their nonlinear dynamics they make an excellent testbed for control algorithms. Their vertical take-off and landing capability makes them attractive for several

* email: pjgarcia@etsii.upm.es

**email: gaurav,monty@robotics.usc.edu

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE MAY 2001		2. REPORT TYPE		3. DATES COVERED 00-00-2001 to 00-00-2001	
4. TITLE AND SUBTITLE Towards vision-based safe landing for an autonomous helicopter				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Southern California, Department of Computer Science, Robotic Embedded Systems Laboratory, Los Angeles, CA, 90089				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT Autonomous landing is a challenging problem for aerial robots. An autonomous landing manoeuvre depends largely on two capabilities: the decision of where to land and the generation of control signals to guide the vehicle to a safe landing. We focus on the first capability here by presenting a strategy and an underlying fast algorithm as the computer vision basis to make a safe landing decision. The experimental results obtained from real test flights on a helicopter testbed demonstrate the robustness of the approach under widely different light, altitude and background texture conditions, as well as its feasibility for limited-performance embedded computers.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 16	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

applications, especially in urban areas. With recent advances in camera technology and image processing hardware, it is natural to consider a vision-based solution to the problem safe landing. To do this, a nadir-pointing camera and an image processing machine are needed. Traditional vision-based approaches to the problem of autonomous landing often rely on a high-contrast landmark, that can be easily identified by standard image processing algorithms [9]. Nevertheless, for a landing strategy to be feasible in real-environment applications, the dependence on a fixed landing mark is limiting and therefore a flexible means of finding safe places to land is required. A challenging algorithmic problem arises though: the need to deal with all kinds of visual terrain features and natural or artificial obstacles that the vehicle may fly over in a real mission. In addition, real time performance, that must be within reach of a small embedded computer, is needed to track the chosen place and steer the helicopter towards it.

This paper presents an algorithm for the visual detection of safe-landing locations. The algorithm is not constrained to any particular outdoor conditions. Therefore assumptions and theoretical considerations are made from a general view point, while mostly real experiments concentrating on a particular set of scenes are used to validate them. As a result, a strategy and an algorithm relying on image processing techniques to search the ground for a safe place to land is presented.

The key contributions of this paper are:

- (1) A computer vision-based approach for autonomous landing in an unknown environment.
- (2) An experimental evaluation of the approach under different flight conditions.
- (3) Four different strategies for safe-landing site detection.

The rest of the paper is organized as follows. The next section discusses some of the most closely related previous work. Then the general description of the approach and the underlying assumptions are presented, followed by the details of the image processing algorithm. We focus on the image description, since such a description is used to look for the landing area. The final sections address the experimental results and the conclusions, along with the intended future work.

2 Related Work

There are several remarkable achievements within the field of autonomous helicopters, where a wide range of research purposes and applications is found.

Three representative samples are found in [6],[13] and [8]. In the first two cases the prototypes are able to fly autonomously in hover and along elementary trajectories, whereas the latter, in addition to autonomous flight, has demonstrated recognition and localization of ground-based targets. In the field of application-oriented prototypes, an unmanned helicopter for monitoring road and traffic networks [3] is also under development, and elsewhere the prototype is intended to perform autonomous inspections of overhead high-voltage power lines [2].

Concentrating on the problem addressed in this research, we are unaware of a vision-based safe-landing system for an autonomous aerial robot. However there are several related achievements that are relevant, since they have certain common aspects with this work. In these approaches the ground is searched from the air and certain features are analyzed for mostly localization purposes.

The visual odometer [1], supported by inexpensive angular sensors to disambiguate rotation from translation, provides a relatively accurate position measure. The customized image processing hardware tracks pairs of target templates in each of the images provided by two ground-pointing cameras and matches them. The system strongly relies on two assumptions: the ground is flat and rich in features. With flat ground, the system must match just one of the targets, and by having certain features, matching and tracking both become more accurate.

Other work in the area of autonomous robotic helicopters was done by the team to which the authors of this paper belong, using the same platform as the one used in the experiments reported here. The experiment described in [12] shows a simple method for estimating the position, heading and altitude of an aerial robot by tracking the image of a communicating GPS-localized ground robot.

In related research dense 3-D surfaces are generated and salient features are extracted and matched to a coarse model of the terrain [5]. This approach is intended to provide autonomous vehicles with a robust absolute position estimation in natural environments, and the prospective applications include self-localization of autonomous aerial vehicles and precision landing on comets and asteroids.

3 General Description of the Approach

3.1 Vision-based landing in the AVATAR project

The USC AVATAR [7] (Autonomous Vehicle Aerial Tracking And Reconnaissance) helicopter (Fig. 1) is a gas-powered model helicopter fitted with PC104 computers and custom control electronics. It has been developed through three generations over the last ten years in our lab. It carries a high-quality Inertial Measurement Unit (IMU), a Novatel RT20 GPS receiver/decoder, an engine RPM sensor and a color video camera. Communication with ground workstations is via 2.4 GHz wireless Ethernet and 1.8 GHz wireless video. The AVATAR control system is implemented using a hierarchical behavior-based control system architecture, which splits the control problem into a set of loosely coupled computing modules called “behaviors”. Up-to-date achievements include autonomous servoing to GPS locations, vision-based tracking and following of moving ground objects, and interaction with ground robots forming a heterogeneous robot group capable of cooperatively performing complex missions [10].



Fig. 1. The USC AVATAR (Autonomous Vehicle Aerial Tracking And Reconnaissance)

We are parallelly addressing autonomous landing by using a high-contrast landmark. Preliminary experiments showing how such a landmark can be found, and its location in the image used, to dynamically guide the helicopter to the landmark are in progress. These will be described in a forthcoming paper.

The research described in this paper is similar to the previous one with respect to the means and the model-free approach but addresses a quite different problem: the safe landing place is no longer provided but has to be found.

3.2 *What is a safe area?*

As a basic starting point we need to define what is considered as a safe place to land. In our approach a safe place is any area on the ground fulfilling the following two conditions:

- (1) The area is big enough for the helicopter to land within.
- (2) The area is essentially clear, i.e., free of either natural or artificial obstacles.

The first condition is a natural one, the robot can not land on a space which is not large enough to accommodate its main rotor and tail boom. Without loss of generality, we look for a circular area of a certain minimum size on which to land.

The second condition needs some explanation. It is necessary but not sufficient for the most general case of autonomous landing. Clearly landing in an area near (or on) obstacles is undesirable. However in the general case, the algorithm should also guide the robot to a safe landing area where the local curvature of the ground is zero and where the local ground plane is horizontal. In this paper we assume that the underlying terrain is flat and horizontal but strewn with obstacles (such as cars, boxes, people, rocks, etc.). Thus we explore the case where the algorithm needs to look for an essentially clear area.

3.3 *Assumptions*

We make the following assumptions:

- (1) The camera is mounted perpendicular to the plane of the ground, pointing straight down.
- (2) The vertical axis of the camera image plane is aligned with the principal axis of the helicopter.
- (3) The image shows a higher contrast at obstacle boundaries compared to the boundaries of visual features due to the terrain texture.
- (4) As already stated above, the underlying terrain is flat.

The first two assumptions affect the performance of the motion-based search strategies that are presented below. Both of these are violated to a small extent by our AVATAR testbed, but our algorithm is robust to such small violations. The third one is a fundamental assumption, as it provides the basis for the algorithm to decide if a visual feature is an obstacle or not. This issue is discussed further in the following sections.

3.4 General strategy

Under the assumption that contrast is higher at the boundaries of the obstacles than elsewhere, the image processing strategy to apply is obvious: a *contrast threshold* must be worked out to provide a criterion of what must be considered an obstacle and what must not. The algorithm must then search the image for a circular area in which all the pixels have a level of contrast below the threshold.

Where and how to look for such an area in the image is an important issue, which we tested using the following four strategies:

- (1) *Single-frame analysis*. The image sequence is analyzed on a single-frame basis, i.e., without taking any advantage of motion and the expected continuity between consecutive frames. The algorithm attempts to find a safe-landing location as close to the center of the image frame as possible.
- (2) *Multi-frame tracking analysis*. Once the search, led by the previous (single-frame) strategy, provides an initial estimate of a safe landing area, such an area is tracked over the rest of the sequence.
- (3) *Multi-frame velocity-vector-based analysis*. Despite the chosen name, the sequence is actually analyzed on a single-frame basis but the search concentrates on the area from which the optical flow enters the image, i.e., on the area right in front of the helicopter. This is done by calculating the relative direction between the helicopter velocity vector (provided by the IMU) and the main axes of the camera image plane (taking into consideration assumption 2 from the previous section).
- (4) *Whole multi-frame analysis*. This approach brings together the previous two multi-frame strategies. The latter provides the first safe area and the other tracks it over the next images.

The results shown in Section 5 show how each succeeding strategy improves upon the performance of the previous one.

4 Image Processing Algorithm

To work out the suitable contrast threshold referred above, two sets of strategies are conceivable:

- (1) An *adaptive threshold based on local analysis*. The image is split into small blocks and a different contrast threshold is worked out for each one according to its local properties. The main advantage of this approach is obvious: the thresholds are expected to provide a good fit to the local

properties of the ground. However, difficulty dealing with different scales of texture (due to the different altitudes at which the helicopter flies during landing) and computing the proper size of the blocks, along with relatively heavy processing, are the discouraging aspects of this approach.

- (2) A *uniform threshold based on global analysis*. A unique and uniform threshold is worked out for the whole image. Such a threshold is obviously expected to fit the local properties more roughly than the adaptive threshold does but, robustness under changing light conditions and different scales of texture and lighter processing make this strategy attractive.

According to these constraints the latter strategy was adopted. The results shown in Section 5 demonstrate why this choice is appropriate.

Two issues have yet to be addressed. First, to choose a proper global *contrast descriptor*. Such a descriptor must be correlated with the “optimum” contrast threshold, i.e., the one which retains just the actual boundaries of the obstacles in the image to deal with and removes spurious boundaries and ground texture features. Second, the *correlation function* between the descriptor and the threshold must be found.

4.1 Contrast descriptor and correlation function

Contrast is generally defined as a measure of the change in the intensity level that pixels display throughout the image. A numeric expression for contrast is given below:

$$C = \sqrt{\frac{1}{NM} \sum_{i,j}^{N,M} (f_{ij} - B)^2}$$

where f_{ij} stands for the value of the pixel located at coordinates (i, j) in an $N \times M$ image having an average brightness level B .

This is a very low-level way of expressing contrast and lacks invariant behavior under changing scale or light conditions. Therefore other higher-level and more meaningful contrast descriptors are needed for our purpose. After trying different options and assessing their correlation with the optimum threshold, we chose as the suitable contrast descriptor the *normalized average of the normalized edge-image histogram*. This dimensionless descriptor, $\frac{\mu}{\sigma}$, is derived by normalizing the edge image and calculating the average (μ) and the standard deviation (σ) of its histogram.

This descriptor can be demonstrated to fulfill the following properties [4]:

- (1) It is theoretically invariant under proportional light changes: $f'_{ij} = k f_{ij}$
- (2) It is theoretically invariant under uniform light changes: $f'_{ij} = f_{ij} + k$
- (3) It is theoretically scale-invariant under any periodic texture patterns, which encourages one to believe that it may also be insensitive under other more realistic textures to a great extent.

To assess how such a descriptor correlates with the optimum contrast threshold the following experiment was performed: among the images collected over a few test flights, a number of them (34), as different as possible, were chosen to make up a meaningful experimental set. All those images have a common pattern: the scene is made up of a grassy background with artificial obstacles (basically boxes) on it. Then a subjective¹ optimum threshold was worked out for each one, along with the contrast descriptor $\frac{\mu}{\sigma}$. Such an optimum threshold was expressed in terms of the percentile of pixels having a higher level of contrast². The pairs of values obtained are shown in Fig. 2.

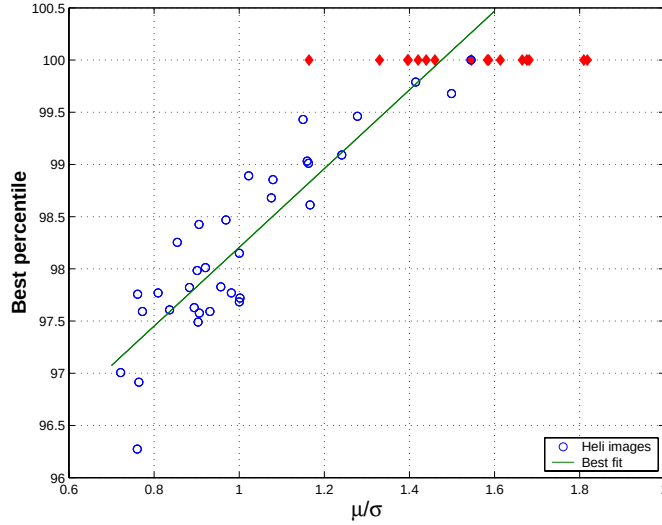


Fig. 2. Pairs of values $\frac{\mu}{\sigma}$ –*Best threshold/percentile* obtained from the experimental set. The least squares linear fit is also shown.

The graph demonstrates that a correlation exists between the descriptor and the optimum threshold. After several fitting trials, a linear fit proved to be the best according to the least squares criterion. Fig. 2 also shows such a fit, whose numeric expression is given below:

$$p = 94.4333 + 3.7724 \frac{\mu}{\sigma} \quad (1)$$

¹ The criterion is actually objective to a certain extent: the optimum threshold is the one which removes spurious features from the image but leaves the boundaries of the obstacles in it. However that occurs in a progressive way along a range (more or less narrow) of threshold values, so there is some room for subjectivity within that range.

² The pixel value in the edge image was taken as the contrast measure.

where p stands for the percentile referred to above.

It may be noted that the fit explains the main cause of variation of the optimum threshold but there are other sources of influence that it does not explain. As a matter of fact, no descriptor based on one-dimensional image representations, like histograms, can determine the optimum threshold. We could expect to do so by using descriptors that pay attention to the two-dimensional spatial relations among the outstanding features, like entropy or moment-based descriptors [11], but such parameters all demand more computation than our application can presently afford.

4.2 Changing scenery: robustness

In order to evaluate to what extent our approach works when the background scenery (the ground texture) changes, the same experiment was performed on a set of collected textures, shown in Fig. 3. Fig. 4 displays the obtained values of the descriptor $\frac{\mu}{\sigma}$ along with those already shown in Fig. 2.

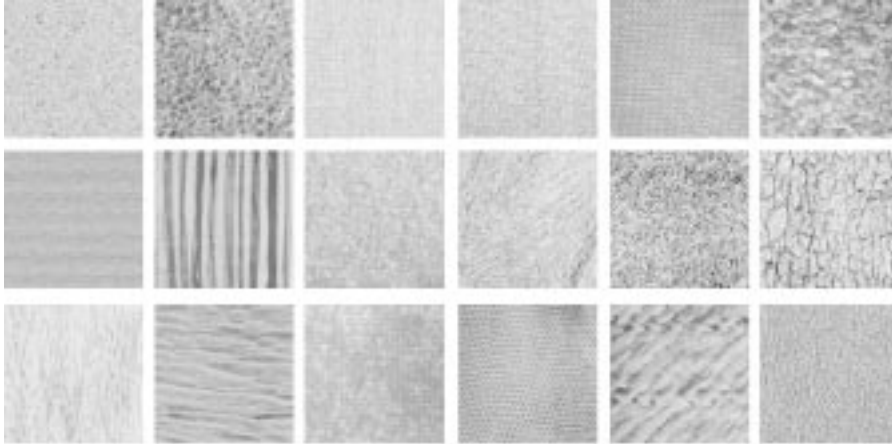


Fig. 3. Set of collected natural textures

Two major facts emerge from this figure: on the one hand, the $\frac{\mu}{\sigma}$ values obtained from the natural textures are relatively close to each other, especially when one takes into account how different those textures are from the point of view of human perception; on the other hand, the range of values they are spread over is narrower than that of the images of the first experimental set (images gathered from the helicopter), all of which share a common background texture (grass)³. In other words, when something different (an obstacle) breaks into a plain and mostly regular texture, the descriptor $\frac{\mu}{\sigma}$ reduces and, in addition, such a reduction is steeper than that we find when the

³ The standard deviation of the descriptor values for the set of textures was 11.53% of their mean, while that of the descriptor values for the aerial image data was 20.63% of their mean.

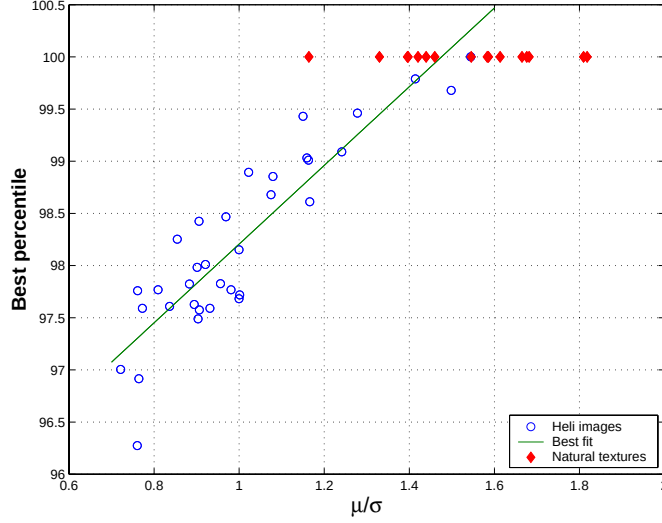


Fig. 4. Values of the texture descriptor $\frac{\mu}{\sigma}$ obtained from the set of natural textures. Just these values matter because the corresponding percentiles are all 100, since they are pure textures without any obstacles within them

background texture is turned into a quite different one. Therefore we are not concerned about what kind of scenery the helicopter is flying over, because the sensitivity of the $\frac{\mu}{\sigma}$ descriptor is negligible compared to when an obstacle intrudes into the field of view. This is the key fact that allows us to assert that this approach is valid, robust and, as shown in Section 5, works well for the intended purpose.

As already mentioned, *scale sensitivity* is also a major concern in our approach. To evaluate to what extent the $\frac{\mu}{\sigma}$ descriptor is scale-invariant a massive experiment was done: several thousand randomly-generated artificial textures at five different scales made up the experimental set. The contrast descriptor was calculated for every one and its scaled versions. Fig. 5 (where only the values obtained for 200 textures are displayed for clarity) shows the results.

The deviation in $\frac{\mu}{\sigma}$ due to scale changes turned out to be 8.26% of its average value, whereas the mean deviation over the textures themselves was 5.30%. These figures point out two important and encouraging facts:

- (1) Texture is again demonstrated to cause only slight changes in the descriptor $\frac{\mu}{\sigma}$, since the mean deviation due to texture itself (5.30%) is even lower than that obtained on the small set of natural textures.
- (2) The descriptor $\frac{\mu}{\sigma}$ is also highly insensitive to scale. Consider that artificially-generated textures differ from natural ones and their discrete and highly changing features make them theoretically⁴ poor candidates for scale invariance. We are thus further encouraged in concluding that $\frac{\mu}{\sigma}$ is scale-

⁴ The basic condition is that the pattern must be continuous, then discretization has some disturbing effects.

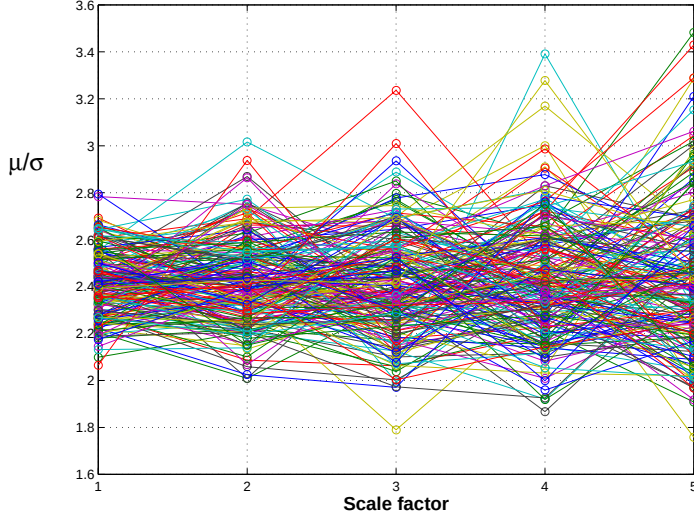


Fig. 5. A sample of the values of $\frac{\mu}{\sigma}$ obtained on 2000 randomly-generated artificial textures

invariant.

5 Results and Discussion

We performed a total of ten real test flights to evaluate the four strategies presented in Section 3.4. We measured how they are affected by different flight conditions. Two conditions were tested: *flight mode* (hovering or motion) and *altitude mode* (a relatively low altitude (~ 10 m) and a higher one (~ 18 m)). A total of 385 images were collected during 2 hours of experiments. Fig. 6 displays two samples of the images gathered during those experiments.

5.1 Offline experiment

The four strategies were tested on the sequences gathered and the following performance indicators were worked out for each one:

- (1) A *change histogram*, showing how much the suggested safe-landing area moves in the image (in pixels) between every frame and the next one and, therefore, the extent to which the same safe area is chosen over the sequence.
- (2) The *image processing rate* (in frames per second)⁵.
- (3) The *failure rate*, i.e., the percentage of images for which the algorithm fails to find a safe landing area. It may be noted that this rate includes

⁵ On a Pentium II@233MHz-CPU machine



(a) Helicopter flying at low altitude (~ 10 m)

(b) Helicopter flying at high altitude (~ 18 m)

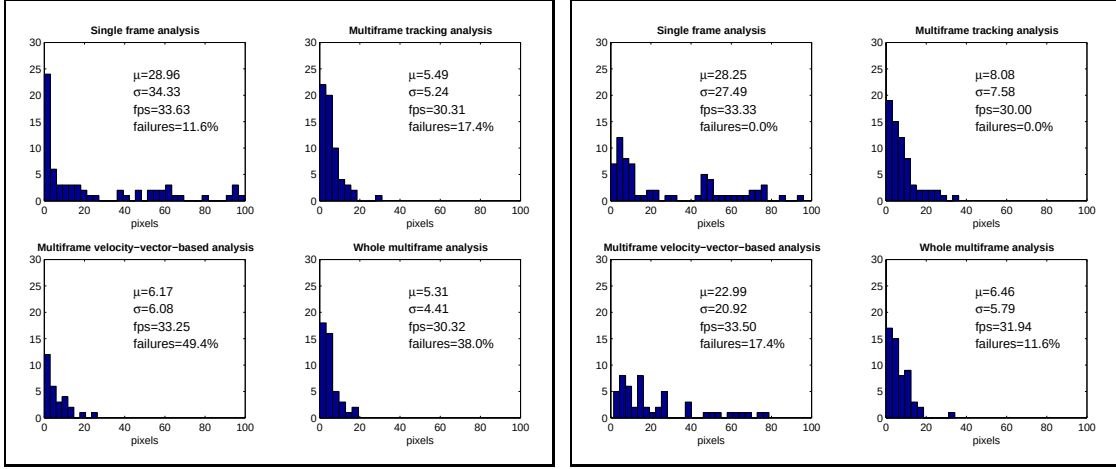
Fig. 6. Two samples of experimental aerial images. The circles stand for the landing area determined by the algorithm and the white rectangles are obstacles (boxes)

both kinds of failure causes: either there is no safe landing place in the field of view or there is actually one but the algorithm did not succeed in finding it.

Another meaningful indicator might be the *false positive rate*, i.e., the percentage of images for which the algorithm makes a mistake and suggests a landing area that is not actually safe (because there are obstacles within it). This will not be discussed further because a quick count revealed it to be negligible ($< 2\%$) in all the strategies. In addition the false positives were only temporary: one or two images later the algorithm no longer considered such an area as safe.

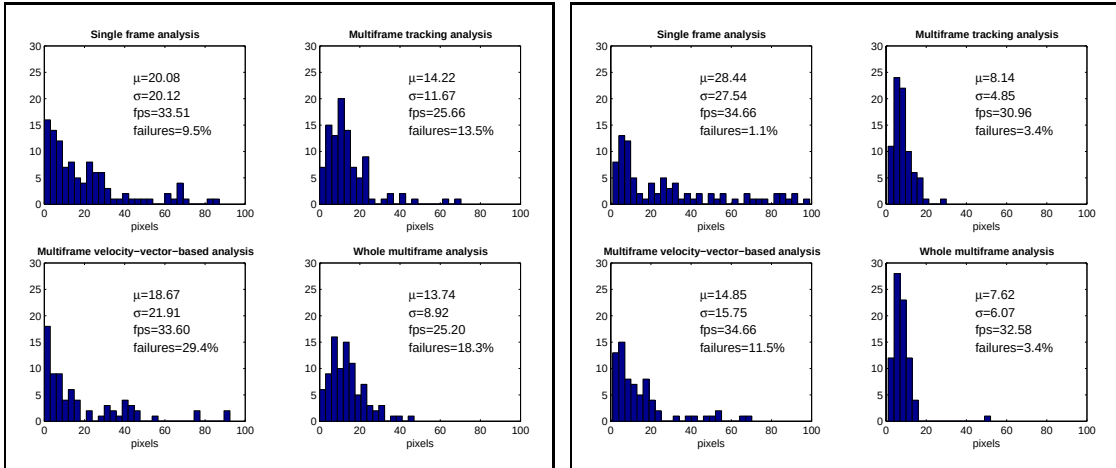
The results are shown in Fig. 7, from which the following inferences may be drawn:

- (1) The higher the altitude, the higher the processing rate and the lower the failure rate. As an example compare the whole multi-frame analysis case shown in Fig. 7a and Fig. 7b. The failure rate in the former is 38% compared to 11.6% in the latter. The processing rate in the former is 30.32 frames per second compared to 31.94 in the latter. This was an expected result taking into consideration that at low altitudes the obstacles occupy relatively larger areas of the image, whereas the required size of the landing circle (in pixels) becomes larger, so the algorithm has much more difficulty finding a large enough area.
- (2) The tracking analysis dramatically improves the continuity, as expected. Even more remarkable is the fact that this is done at a very moderate computational cost. This is apparent when the μ values are compared between the single frame and multi-frame analysis for any of the altitude or motion conditions. As an example consider Fig. 7d. For the single



(a) Helicopter in hover at low altitude (~10 m)

(b) Helicopter in hover at high altitude (~18 m)



(c) Helicopter in motion at low altitude (~10 m)

(d) Helicopter in motion at high altitude (~18 m)

Fig. 7. Performance of the different search strategies under different flight modes and altitude conditions. Each figure is a four-histogram set showing the change in suggested safe-landing area (in pixels) from frame to frame for each search strategy.

frame analysis $\mu = 28.44$, while for the multi-frame analysis $\mu = 8.14$. This improvement is achieved at the cost of less than 4 frames per second in processing speed.

- (3) Letting the velocity vector lead the search is obviously effective. This can be seen by comparing the results of the pure tracking and the whole multi-frame strategies. In spite of the rise in failure rate (a expected result taking into consideration that the velocity-vector-strategy forces the algorithm to find the landing area in a restricted part of the image), the rest of the figures improve. On the one hand, continuity becomes

higher because of the fact that the first detected safe area has just entered the image, so it is expected to take longer until leaving it and, therefore, forcing the algorithm to look for a new one; as a consequence, processing rate also rises slightly because the algorithm has to look for a new landing place fewer times.

- (4) The *whole multi-frame* strategy is definitely the best. There is evidence for this from the indicators shown in Fig. 7 and also from observing the separate influence of the two strategies fused in it: tracking and search led by the velocity vector.

In short, the figures suggest that when the search is performed at relatively high altitudes (as it seems reasonable to do) the tracking strategy provides the expected continuity between consecutive frames at an affordable computational cost, while the velocity-vector-based approach speeds up processing, along with making the landing manoeuvre easier for the helicopter since the landing area is in front of it. Finally, the whole multi-frame strategy, by fusing both approaches, takes advantage of both and stands out as the best approach.

6 Conclusions and Future Work

We have experimentally demonstrated a straightforward and inexpensive description of aerial imagery that can be used to find a safe landing area. Such a capability is essential for an autonomous aerial robot. Our strategy is robust and lightweight. The results show how a set of non-coupled strategies can be combined to assist a robot helicopter to perform an autonomous landing. In addition, the high processing rates suggest that the approach is feasible and the algorithm is implementable in limited-performance embedded computers.

Future work includes tests over terrain with rougher and irregular texture patterns and features, as well as a further development of the link between the vision and the control systems. The latter is expected to be accomplished with a straightforward strategy: once the landing site has been found, the vision system will take over the guiding responsibility by sending two-component velocity commands to the control, in order to reduce the offset between the center of the landing area and that of the image.

Moreover, we foresee a further challenge: a whole autonomous landing experiment, in which the manoeuvres, from the moment at which the helicopter begins to look for the landing area until touch down, will all be autonomous. This prospective achievement will require new vision-control collaboration features to be implemented, including a landing abort mechanism. This capability will prevent the helicopter from landing when the image processing system detects obstacles within the landing area that had not been detected at larger

scales (higher altitudes).

7 Acknowledgments

The authors thank the USC AVATAR team, whose readiness and hard work made the experiments possible. This work is supported in part by DARPA grant DABT63-99-1-0015 under the *Mobile Autonomous Robot Software* (MARS) program and DARPA contract F04701-97-C-0021 under the *Tactical Mobile Robotics* (TMR) program. The financial support for the first author's work, given by the Ministry of Education of Spain and the Department of Systems Engineering and Automatics at Universidad Politecnica Madrid (Spain), is also gratefully acknowledged.

References

- [1] O. Amidi, T. Kanade, and K. Fujita. A visual odometer for autonomous helicopter flight. *Robotics and Autonomous Systems*, 28:185–193, 1999.
- [2] P. Campoy, A. Barrientos, P. J. Garcia-Pardo, J. del Cerro, and I. Aguirre. An autonomous helicopter guided by computer vision for visual inspection of overhead power cable. In *Proceedings of the 5th International Conference on Live Maintenance*, Madrid, May 2000.
- [3] P. Doherty, G. Granlund, K. Kuchcinski, E. Sandewall, K. Nordberg, E. Skarman, and J. Wiklund. The WITAS unmanned aerial vehicle project. In *Proceedings of the 14th European Conference on Artificial Intelligence*, Berlin, August 2000.
- [4] P. J. Garcia-Pardo. On the properties of a contrast descriptor for aerial imagery. Technical Report IRIS-00-391, Institute for Robotics and Intelligent Systems, University of Southern California, November 2000.
- [5] A. E. Johnson. Surface landmark selection and matching in natural terrain. In *Proceedings of the IEEE Computer Vision and Pattern Recognition*, June 2000.
- [6] T. J. Koo, D. H. Shim, O. Shakernia, B. Sinopoli, Y. Ma, F. Hoffmann, and S. Sastry. Hierarchical hybrid system design on Berkeley UAV. In *International Aerial Robotics Competition*, Richland, Washington, August 1998.
- [7] J. F. Montgomery. The USC Autonomous Flying Vehicle Project (AFV) Project: Year 2000 Status. Technical Report IRIS-00-390, Institute for Robotics and Intelligent Systems, University of Southern California, September 2000.
- [8] M. Musial. MARVIN: An autonomously operating flying robot. Project homepage [online]: <http://pdv.cs.tu-berlin.de/MARVIN>, last update 2000.

- [9] C. S. Sharp, O. Shakernia, and S. S. Sastry. A vision system for landing an unmanned aerial vehicle. Submitted to *2001 IEEE International Conference on Robotics and Automation*, 2001.
- [10] G. S. Sukhatme and J. F. Montgomery. Heterogeneous robot group control and applications. In *International Conference of the Association for Unmanned Vehicle Systems*, Baltimore, Maryland, 1999.
- [11] C. H. Teh and R. T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10:496–513, 1988.
- [12] R. T. Vaughan, G. S. Sukhatme, F. J. Mesa-Martinez, and J. F. Montgomery. Fly spy: Lightweight localization and target tracking for cooperating air and ground robots. *Distributed Autonomous Robotic Systems*, 4:315–324, 2000.
- [13] B. Woodley, H. Jones II, E. LeMaster, and E. Frew. Carrier phase GPS and computer vision for control of an autonomous helicopter. In *ION GPS-96*, Kansas City, Missouri, September 1996.