Learning to Detect Aircraft at Low Resolutions

Stavros Petridis*, Christopher Geyer, and Sanjiv Singh

Robotics Institute, Carnegie Mellon University,
Pittsburgh, PA, USA
sp104@doc.ic.ac.uk, {cgeyer,ssingh}@cs.cmu.edu

Abstract. An application of the Viola and Jones object detector to the problem of aircraft detection is presented. This approach is based on machine learning rather than morphological filtering which was mainly used in previous works. Aircraft detection using computer vision methods is a challenging problem since target aircraft can vary from subpixels to a few pixels in size and the background can be heavily cluttered. Such a system can be a part of a collision avoidance system to warn the pilots of potential collisions. Initial results suggest that this (static) approach on a frame to frame basis achieves a detection rate of about 80% and a false positive rate which is comparable with other approaches that use morphological filtering followed by a tracking stage. The system was evaluated on over 15000 frames which were extracted from real video sequences recorded by NASA and has the potential of real time performance.

Keywords: automatic target detection, aircraft detection, collision avoidance, obstacle detection, computer vision applications.

1 Introduction

The use of computer vision for detecting obstacles in the flight path of an aircraft is investigated in this paper. Such a system can be used to warn the pilots for potential collisions and would also be useful aboard unmanned aerial vehicles (UAVs). Any aircraft detection technique should provide high detection rate, low false alarm rate and early detection. The main challenges in this problem are the presence of image noise, the almost stationary nature of the target on collision course, the possible presence of heavily cluttered background, and the extremely small size of the obstacles that can vary from subpixels to a few pixels. In addition, the algorithm should be able to run in real time imposing severe constraints on execution time.

Obstacle detection in the flight path of an aircraft is a key component of a collision avoidance system. That also includes a collision risk estimation component together with a component that performs appropriate avoidance maneuvers in order to maintain minimum separation distances. Such an approach must demonstrate a level of performance which meets or exceeds that of a human pilot as stated in FAA order 7610.4 [1].

* This work was done while the author was visiting the Robotics Institute at Carnegie Mellon University. He is currently with the Department Of Computing, Imperial College London.

A. Gasteratos, M. Vincze, and J.K. Tsotsos (Eds.): ICVS 2008, LNCS 5008, pp. 474–483, 2008.

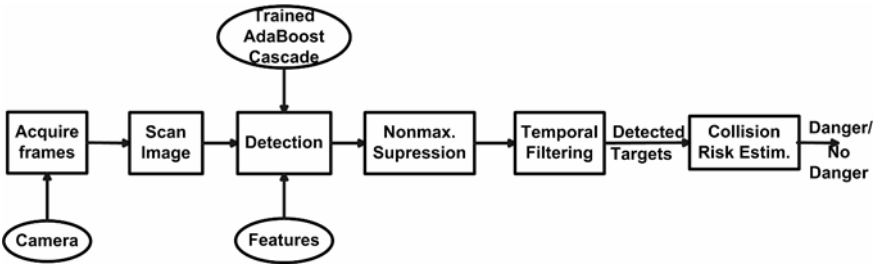

Fig. 1. The proposed architecture for collision detection

There are several different approaches which address the problem of obstacle detection in the framework of collision avoidance including morphological filtering [2], dynamic programming [3], and recursive max filter [4]. McCandless [5] proposed an optical flow method which is suitable only for moving objects. Gandhi [6] proposed a two stage approach, an image processing stage followed by a tracking stage. The image processing stage isolates potential features using morphological filtering and the tracking stage tracks these features to distinguish the real targets from background clutter using the rate of translation and expansion. Carnie [7] implemented a similar approach using morphological filtering followed by a dynamic programming algorithm to enhance detection.

The use of morphological filtering is popular on computer vision based collision avoidance systems [2], [6], [7]. However, this approach generates a significant number of false positives and requires tracking of the features over a large number of frames as reported in [7], [8].

In this paper we present a machine learning approach based on AdaBoost to detect aircraft. We report results on the application of a modified version of the Viola and Jones object detector [9], which has been used successfully in face detection, on the aircraft detection problem. Initial experiments show that it achieves a high detection rate (around 80%) whereas the number of false positives generated by the static classifier, i.e. no tracking is used, is comparable with that obtained by systems based on morphological filtering and tracking. The system was evaluated on over 15000 frames taken from real video sequences which is a much higher number than the typical number of test frames used by the majority of the existing aircraft detectors (less than 1000). An attractive property of this detector is that it has the potential to run on or close to real time. The original Viola and Jones classifier tries to detect faces from their internal structure whereas our classifier tries to separate the target from the background since targets which are far away do not have any structure; they appear as a row or a rectangle of few pixels.

The paper is organized as follows: section 2 gives an overview of the proposed system, section 3 describes the data we used, section 4 discusses the results of two different scenarios (collision course and crossing) and section 5 concludes the paper.

2 System Overview

The system we propose here is based on the framework introduced by Viola and Jones [9]. It consists of six stages as shown in Fig. 1. In the first stage an image frame is acquired by the camera and then in stage two a sliding window is used to scan the frame. The next 3 stages are the main parts of the system and they are consisted of the detector followed by a non-maximal suppression stage which is followed by a temporal filtering stage. Then all the detected targets (both real targets and false positives) are passed to the final stage which determines which targets pose a threat to the aircraft. The main focus of this paper was the development of stage 3 but some simple temporal filtering approaches were evaluated. The system shown in Fig. 1 should work in real time. However, the detector should be trained off – line and when the training process ends then it can be used on line as well.

2.1 Cascade Training

A cascade is a structure which consists of a series of classifiers as shown in Fig. 2. When a pattern is fed into the cascade then it is labeled as positive if it successfully passes all stages whereas there are multiple exits for the patterns which fail at some point and those are labeled as negative.

In the third stage, a cascade classifier is applied to all subwindows of each frame. The key point is that within any single frame the vast majority of subwindows are negative so the cascade attempts to reject as many negatives as possible at the earliest stage possible. So the first stages consist of simple and fast classifiers which reject most of the negative subwindows while keeping almost all of the positive subwindows whereas subsequent stages eliminate additional negatives but they are more complex. Stages in the cascade are constructed by training classifiers using AdaBoost as in [9]. However we do not use the original cascade where each classifier outputs "Yes" or "No" but we follow the boosting chain approach [10] where each classifier outputs a real value which is added to the sum of the outputs of all the previous classifiers. If the new sum is greater than the stage's threshold then the pattern is fed into the next classifier, otherwise it is discarded. So the cascade's function can be summarized in the following three steps

1. Given an input pattern initialize $s = 0$
2. For all stages ($i = 1 \ldots K$)
   $s = s +$ stage output (c_i)
   if $s <$ stage threshold (r_i) then exit with negative response
3. Exit with positive response

The output value of the i-th stage is c_i and the corresponding rejection threshold is r_i (see Fig. 2). A real example of how the cascade works is shown in Fig. 2 (right). The X-axis represents the stages of the cascade and the Y-axis represents the cumulative sum of the stage outputs. Similarly to [11] we can define a rejection trace (black trace) which represents the thresholds of each stage. So as long as the partial sums are greater than the rejection trace then the pattern is kept. Blue lines represent aircraft (targets) and red lines represent negative patterns. So the red lines that are above the rejection threshold until the final stage are false positives. The red dashed lines

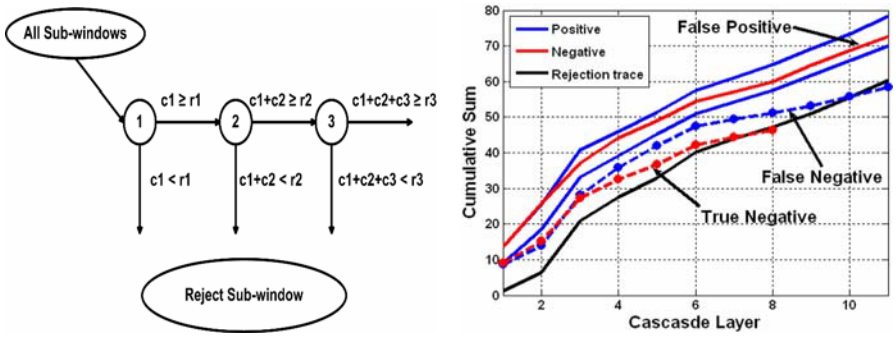

Fig. 2. Left: The cascade structure used in this paper, Right: An example of how the cascade works

represent rejected negative subwindows whereas the blue dashed-lines represent false negatives. Using this approach the performance of a pattern in the prior stages is taken into account and also a pattern may fail one or more stages but as long as it stays above the rejection threshold it can be correctly classified.

2.2 Features

The 4 types of Haar features from which the training system will extract the best features are shown in Fig. 4 Apart from the two commonly used filters, two more filters were introduced to take into account the nature of the problem. A target in a large distance is expected to have greater width than height as shown in Fig. 3 (right). Since we can not detect the aircraft from its structure (as in the case of face detection) but we want to distinguish the target from the background then the inclusion of these filters helps towards this goal (Fig 4, top row, columns 3-4). The features are used in all locations and scales that fit in the detector. The output of each feature is the sum of the pixels which lie within the white rectangle subtracted from the sum of the pixels which lie in the black rectangle.

The four best features selected by AdaBoost are shown in the bottom of Fig 4. It is obvious that all of them measure the difference in intensity between the target (mainly rows 6 and 7) and the background. The best feature is of type 3 and the second best feature is of type 4. This fact also justifies the introduction of the new features.

2.3 Non-maximal Suppression and Temporal Filtering

In the fourth stage, non-maximal suppression is performed to the output of the classifier to get a single detection in a 13 x 13 neighborhood. The cascade classifier is insensitive to small changes in translation so multiple detections usually occur near each positively detected pixel.

In the fifth stage, a simple temporal filtering is used. For each detection in frame k we check a 5 x 5 neighborhood around the target in frame k-1. If a detection exists in that neighborhood in the previous frame then the detection in frame k is kept, otherwise it is discarded. We consider the n previous frames for every detection and if

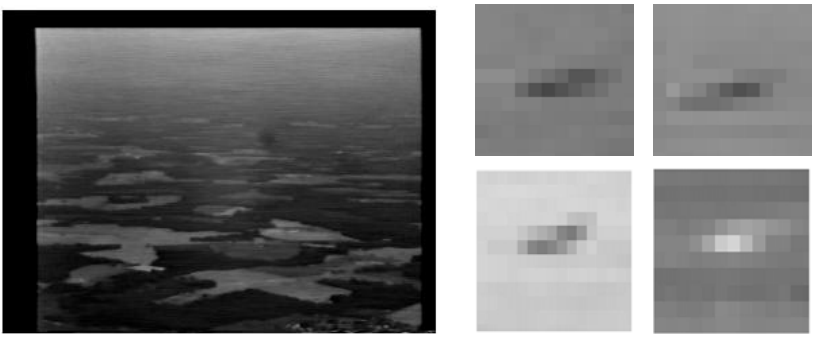

Fig. 3. Left: a heavily cluttered scene from our data, Right: Examples of positive training images

there are at least $m < n$ detections in those n frames then this detection is identified as a target. We chose a 5 x 5 neighborhood since a target on collision course is expected to be almost stationary.

3 Data Collection and Training

We had access to about 4 hours of video that was captured during test flights conducted at NASA Langley Research Center in 1997 [5]. We did not have access either to the navigational data or to the distance between the target and the host aircraft for each frame. In total, there are 15 sequences with aircraft and 3 different classes of maneuvers. There are 4 sequences where the target aircraft is on collision course (~110 sec), i.e. it flies towards the host aircraft, 5 where the target aircraft flies perpendicular to the host aircraft (~22 sec) and 6 where the target aircraft flies directly away from the host aircraft (~ 480 sec). A frame from our data is shown in Fig. 3. We manually extracted all the 15 sequences that contain aircraft (total ~10min). In 8 sequences there is low clutter in the background whereas in the rest the background is heavily cluttered, including clouds, ground and sea regions.

In order to train the cascade we used 1266 positives patterns from 2 of the crossing object maneuvers. For each pattern also its mirror image and its left and right translation by a few pixels were included in the training set in order to generate a less biased training set [12]. All the patterns were contained in a 12 x 12 window. Some examples of the training images are shown in Fig. . Training was performed in exactly the same way as the original cascade detector proposed in [9]. The final detector is an 11 layer cascade of classifiers which includes 335 features. The number of features used by each classifier is: 8, 12, 16, 22, 32, 33, 34, 36, 44, 48 and 50.

The top row of Fig. 3 shows two aircraft which fly perpendicular to the host aircraft and the bottom row shows two aircraft on collision course. Since the appearance of the aircraft in these two cases is quite similar we used the same classifier in both cases.

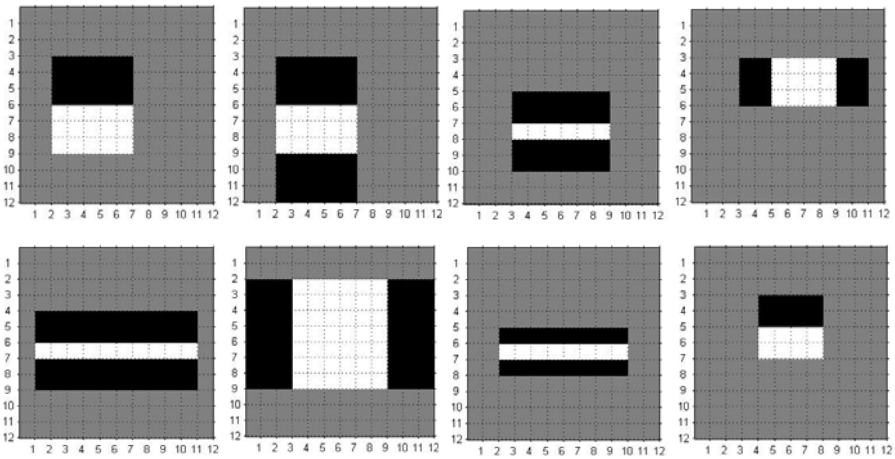

Fig. 4. Top row: The 4 types of features used, Bottom row: The 4 best features selected by AdaBoost for the first classifier

4 Results

4.1 Collision Course Scenario

We tested the trained classifier on the collision course maneuvers and the results are shown in Table 1, and Fig. 5. The background of sequence 1 and 3 is smooth whereas the background of sequences 2 and 4 is heavily cluttered. Fig. 4 (right) shows a frame from sequence 2 together with the classifier's output (squares). So in this frame we see that the plane is correctly detected (top right) but there are also four false positives.

We tested the classifier from the point that the target occupies at least 3-4 pixels since the performance of the classifier for targets less than 4-5 pixels is poor. Fig. 5 shows the ROC curves for the second and fourth sequence respectively. In the first case (sequence 2) the classifier ran for the part of the sequence between 25 seconds to collision and 10 seconds to collision whereas in the second case (sequence 4) the classifier ran for the part of the sequence between 20 seconds to collision and 5 seconds to collision. . The operating point shown in the ROC curves corresponds to the default classifier trained by AdaBoost. Both sequences 2 and 4 are heavily cluttered so the use of the static classifier generates a significant number of false positives per frame, 7.9 and 14.67 respectively. However, the detection rate is very good 84.3% and 89.14% respectively

The use of the temporal filtering approach has a positive effect on the system's performance. After trying a series of different values for m and n we found that the set of values $m = 25$, $n = 50$ result in satisfactory performance. For example, $m = 25$, $n = 50$ means that in order to consider one detection as valid then at least 25 detections should occur in the current detection neighborhood in the last 50 frames. In order to better demonstrate the effect of this temporal filtering approach we removed two

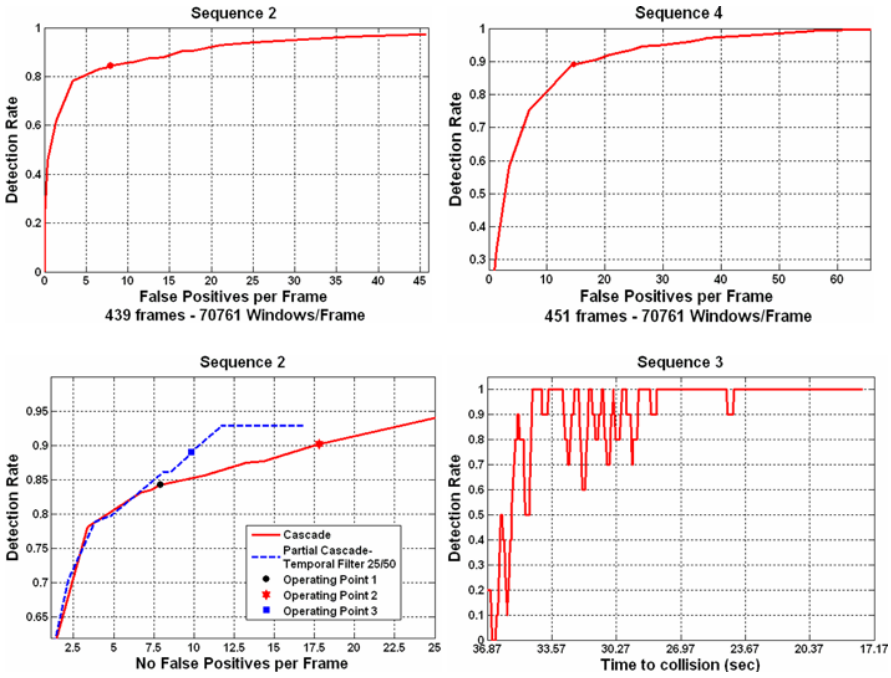

Fig. 5. Top Left: ROC curve for sequence 4, Top Right: ROC curve for sequence 2, Bottom Left: ROC curve for sequence 2 using temporal filtering, Bottom Right: Detection rate as a function of time to collision

layers of the cascade (so 9 classifiers in this case) so a much higher number of false positive detections is generated. The second operating point in Fig. 5 corresponds to the 9-layer cascade and the third operating point corresponds to the 9-layer cascade after the above mentioned temporal filter is applied. It is obvious from Fig. 5 that the application of this simple temporal filtering results in a significant reduction of the number of false positives, about 7.5 false positives per frame.

From Fig. 5 we see that a relatively high number of false positives detections is generated per frame for sequences 2 and 4. Although this is not acceptable for a real world application it is an excellent result from a machine learning point of view. In each frame the classifier checks more than 70000 windows and only 7-15 windows are identified as false positives. This is expected since the classifier detects target-like objects so high cluttered background increases the probability of regions with the same pixel distribution as a real target. The performance of the classifier in the all sequences is shown in Table 1.

Apart from the four collision course sequences there were also six sequences where the target aircraft is flying directly away from the host aircraft. If we play those sequences backwards then it seems that the target aircraft is on collision course. The main drawback of this approach is that the time to collision does not correspond to a real collision course scenario. In such a scenario the relative velocity between the two

Table 1. Detection rate and false positives (FP) for all sequences

	Detection Rate / FP per frame	Frames
Sequence 1	81.68% / 0.005	606
Sequence 2	84.3 % / 7.9	439
Sequence 3	96.38% / 0.003	1107
Sequence 4	89.14 % / 14.67	451
Average	87.88 % / 3.88	-

Table 2. Detection rate and number of FP for the sequences that the target aircraft is flying directly away from the host aircraft

	Detection rate / Total No of FP	Total No of frames
Sequence 5	83.15% / 0	902
Sequence 6	90% / 264	1181
Sequence 7	92.28% / 0	1400
Sequence 8	88.47 % / 115	1301
Sequence 9	87.44% / 0	1401
Sequence 10	85.98% / 663	1270
Average	87.89%	-

Table 3. Detection rate and number of FP for the crossing target maneuvers

	Detection rate / Total number of FP	Frames with planes
Sequence 1C	98.45% / 2	129
Sequence 2C	97.65% / 6	213
Sequence 3C	97.09% / 20	103
Sequence 4C	99.34% / 3	151
Sequence 5C	100% / 0	41
Average	98.506%	-

aircraft is the sum of their velocities. On the other hand, when the aircraft flies directly away then the relative velocity is the difference between the two velocities (in both cases the velocities are collinear). So when we play that video sequence backwards the target aircraft approaches the host aircraft with a much lower speed. This results in an increased time to collision and this is obvious from Table 2 in which the number of frames used is much higher than the number of frames used in the real collision course sequences.

Fig. 5 (bottom right) shows the detection rate as a function of the time to collision. The third sequence from the collision course results section was used in this experiment. In order to get a single point in the curve we average ten consecutive frames. We see that in the beginning the detection rate is low and as the target approaches the host aircraft (time to collision decreases) the detection rate increases. We should also note that the detection rate for about 6-7 seconds is high but the detections are not consistent as shown from the fluctuation around 80%. After a point

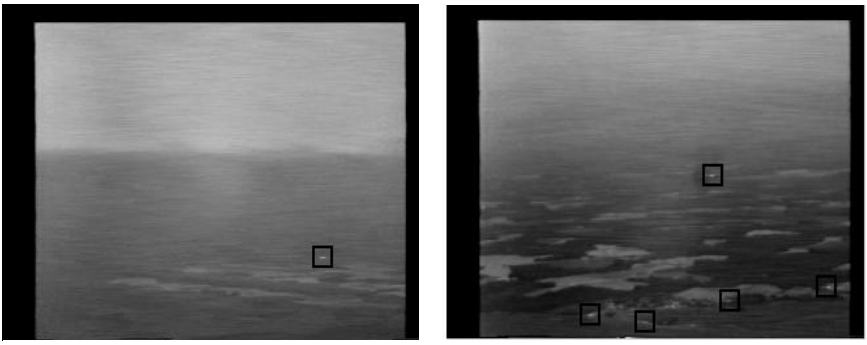

Fig. 6. Left: An aircraft on a crossing maneuver, Right: An aircraft on collision course (top) together with four false positives

the target is detected consistently and consequently the detection rate becomes constant. The background of this sequence is smooth and that so it is reasonable to consider the detection rate obtained as upper limit. In a heavily cluttered sequence or in a sequence with low contrast it is expected that the detection rate will be worse.

4.2 Crossing Scenario

We also tested the trained classifier on the crossing target maneuvers and the results are shown in Table 3. Sequences 1C and 4C were used for training and sequence 2C was used for validation so it is expected a high detection rate and a low false positive rate for those sequences. Fig. (left) shows a frame from sequence 2 in which the plane is successfully detected and there are no false positives.

The remaining sequences (3C and 5C) were used for testing. The background of sequence 5C is almost uniform and that is why the target is detected in all frames with zero false positive alarms. Sequence 3C is heavily cluttered but the performance of the classifier is very good. This may be the result of the similar background and lighting conditions with sequence 4C. We should note here that most of the missed detections occur when the target enters or leaves the scene and therefore the classifier fails to detect a partial target.

5 Conclusions

In this paper a different approach to the problem of aircraft detection was presented. Instead of using an image processing method we used a learning method to learn the targets from real data. The system described here is a very popular method in the area of face detection and with few modifications was successfully used in the problem of target detection in video sequences. The advantage of this method is that it detects the targets on a frame to frame basis with a high detection rate, usually around 80%, and lower false positive rate than the commonly used morphological filtering which is susceptible to false positives. The described approach achieves a false positive rate which usually lies between 1 and 5 false positives per frame, and in some cases up to

10, without further processing and this is an encouraging result. The above results were obtained using over 15000 test frames which is a much higher number than the typical number of frames used by the majority of the detection systems. The described system has also the potential of real time execution.

We were mostly focused on the detection part and developed a detection framework that works on a frame to frame basis. In this work we used very simple temporal filters so future work includes the use of more complicated temporal filters or tracking algorithms which are expected to further enhance the system's performance. The consistent detection of targets which occupy less than 5 pixels is something that has to be addressed too.. Finally the development of a reliable system to estimate the collision risk is an important issue that should be addressed in order to have a final collision avoidance system.

Acknowledgments. We would like to thank Dr J. W. McCandless from NASA Ames Research Center for providing us with the data.

References

1. Federal Aviation Administration, ch. 12, Section 9, Remotely Operated Aircraft, in Order 7610.4: Special Military Operations, US Gov. Printing Office
2. Casasent, D., Ye, A.: Detection filters and algorithm fusion for ATR. IEEE Trans. On Image Processing 6(1), 114–125 (1997)
3. Barniv, Y.: Dynamic programming solution for detecting dim moving targets. IEEE Trans. On Aerospace and Electronic Systems 21(1), 144–156 (1985)
4. Nishiguchi, K., Kobayashi, M., Ichikawa, A.: Small target detection from image sequences using recursive max filter. In: Proc. Of SPIE, pp. 153–166 (1995)
5. McCandless, J.W.: Detection of aircraft in video sequences using a predictive optical flow algorithm. Optical Engineering 3(3), 523–530 (1999)
6. Gandhi, T., Yang, M.T., Kasturi, R., Camps, O., Coraor, L., McCandless, J.: Detection of obstacles in the flight path of an aircraft. IEEE Trans. On Aerospace and Electronic Systems 39(1) (2003)
7. Carnie, R., Walker, R., Corke, P.: Image processing algorithms for UAV Sense and Avoid. In: Proc. ICRA, pp. 2848–2853 (2006)
8. Gandhi, T., Yang, M.T., Kasturi, R., Camps, O., Coraor, L., McCandless, J.: Performance Characterization of the Dynamic Programming Obstacle Detection Algorithm. IEEE Trans. On Image Proces. 15(5), 1202–1214 (2006)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. CVPR 2001, pp. 511–518 (2001)
10. Xiao, R., Zhu, L., Zhang, H.J.: Boosting chain learning for object detection. In: Proc. ICCV, pp. 709–715 (2003)
11. Bourdev, L., Brandt, J.: Robust object detection via soft cascade. In: Proc. CVPR 2005, pp. 236–243 (2005)
12. Sung, K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Trans. On PAMI 20(1), 39–51 (1998)