

Distant Aircraft Detection in Sense-and-Avoid on Kilo-Processor Architectures

Tamas Zsedrovits[†], Akos Zarandy^{**†}, Borbala Pencz^{*}, Antal Hiba^{**†}, Mate Nameth^{**†}, Balint Vanek^{*}

[†]Pazmany Peter Catholic University, Faculty of Information Technology and Bionics, Budapest, Hungary,

^{*}Institute for Computer Science and Control of the Hungarian Academy of Sciences (MTA SZTAKI) Budapest, Hungary

Abstract— In this paper an algorithm for distant aircraft detection for visual sense-and-avoid for UAV is presented. The algorithm uses local edge density to partition the frame into two types of regions. The first type is the unstructured or homogeneous part like sky region and the second part where there is a structured background, like high contrast clouds or terrain regions. The airplanes are detected on the two types of regions with different strategies. The algorithm was planned to run in an embedded environment with low power consumption, thus it can be run onboard of a small or mid-size UAV. First steps towards the GPU implementation on the nVidia Jetson TK1 development board are done and also presented in the paper.

Index Terms — UAV, UAS, Sense-and-Avoid, Detection, Camera, FPGA, GPU

I. INTRODUCTION

The Unmanned Aircraft Systems (UAS) technology is close to provide effective solutions in many applications thanks to the considerable evolution in the last decade. Next to military applications, which are more common especially from the late '90s, nowadays more and more commercial applications and services emerge based on this technology [1], [2], [3].

However, besides the challenges in research and development, the regulations that standardize the integration of UAS into current air traffic is also imperfect. For example, in many applications autonomous flight capabilities would be vital to realize cost effective operation, but it is not allowed due to the lack of satisfaction of the necessary safety requirements, even though many UAS have this ability.

The sense-and-avoid (SAA), or collision avoidance (CA) capability is one of the most important problems which has to be solved [4]. The SAA functions have to be run on-board even if the radio link in between the base station and the aircraft is lost. Otherwise in the case of an autonomous task, UAS would fly “blindly”, only following GPS coordinates.

The integration will be started with smaller aircraft and followed by the larger ones, as it is written in the roadmap for remotely piloted aircraft systems in the European Union [5]. A vision based SAA would be smaller and more cost effective, than other solutions,

like RADAR. The size and the cost is important in many commercial application, especially when a small aircraft is used. The camera based solution has some disadvantages, like weather limitations, but these can be handled with another devices, like infrared cameras, or with safety procedures. The goal was here to make the system work in acceptable weather conditions.

The kilo-processor chips which are available today allow us to implement complex algorithms, with high performance and low power consumption. In the paper the concept of SAA is introduced briefly including the current solution and implementation on kilo-processor chips (FPGA, GPU).

II. SENSE-AND-AVOID

In this section the basic concept of collision avoidance and sense-and-avoid is introduced. As most of the time because of the size and energy limitations, given by the airframe, the sense-and avoid capability is one of the most important feature that UAS must have before their integration into the common airspace.

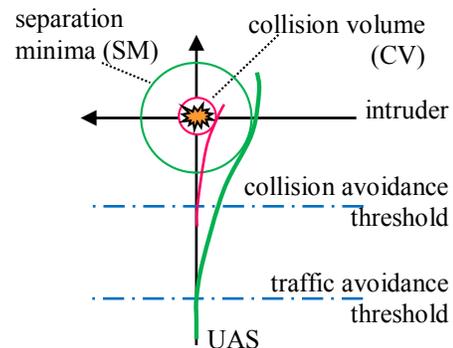


Figure 1. Collision avoidance

In order to reduce the risk of mid-air collisions aircraft have to keep a separation minima (SM) from one another, which has a well-defined value depending on the size of the aircraft. The volume given by the SM can be avoided if the avoidance maneuver is started at the traffic avoidance threshold. On the other hand if the maneuver is started at the collision avoidance threshold, at least the collision can be avoided.

In the air traffic management a layered approach is used for the CA from the rules of the air traffic through

the so called airborne collision avoidance systems (ACAS), which is a cooperative solution, to the non-cooperative sense-and-avoid [6], [7]. In general only bigger and most expensive aircraft are equipped with ACAS, thus for small aircraft the pilot is responsible for the CA and for small UAS the on-board SAA is the only solution.

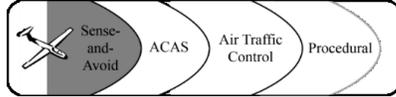


Figure 2. The layered concept collision avoidance

A. Vision based sense-and-avoid system

In the literature there are many approaches to address the SAA problem. One of the best current SAA solutions is introduced in [8]. There, the system uses information from RADAR as well as from electro-optical EO sensor. The main advantage of this system is that it is capable of running the SAA in all-time all-weather conditions. Due to the camera sensor it is more reliable and more accurate than other RADAR systems. The main drawback of the system is the problem caused by the fusion of different sensors. The system cannot be cheap because of the used sensors, and it is heavy as well, so it cannot be used on a mid-size or small UAS.

Another example is shown in [9], where a hidden Markov model (HMM) based temporal filtering is introduced for the detection with the addition of relative bearing and elevation estimation capabilities. The main advantage of that research is that they have access to various types of aircraft, sensors and computational resources. The detection range and false alarm rates are very impressive, and the authors have the biggest known airborne video database as well, with a real target aircraft. The main drawback seems to be the power consumption of the proposed system due to the computationally extensive preprocessing and temporal filtering steps. The power consumption of the presented implementation is more than 59W, which is again too much for a mid-size or small-size UAS.

Fulfilling the size, power and cost restrictions, a closed-loop visual SAA system was developed and described in our previous publications. The vision based collision avoidance system is constructed from a sensor-processor device, which calculates the relative view angle and the angular size of the intruder aircraft, and a software component on the central navigation system of the UAV, which evaluates these data, and commands the aircraft if a maneuver is required.

The block diagram of the vision system is shown in Figure 3. It contains 5 miniature camera with M12 optics, an FPGA board for performing the processing, and an SSD, to collect the raw flight data to do off-line processing for algorithm development and verification

purposes. The hardware of the vision system is described in detail in [10].

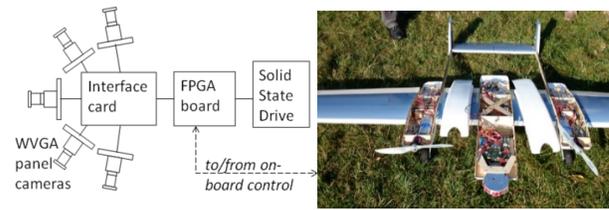


Figure 3. The block diagram of the visual sensor-processor system, and its physical implementation mounted on the nose of a UAV.

The SAA system operates on a way that the vision system detects the intruder aircraft relative angular position to the coordinate system of our UAV. The vision system sends these data towards the GNC system of our UAV. This system is equipped with inertial and navigation sensor (INS), which permanently measures the position, the attitude, and the dynamics of the aircraft. Based on the visual data and the data from the INS the relative position and velocity of the intruder is estimated with an Extended Kalman Filter (EKF), assuming that the intruder is on a straight path with constant velocity. Based on the relative position and velocity, the collision risk is estimated and the autopilot conducts a collision avoidance maneuver if it is needed. The operation of this unit is described in [11].

III. DISTANT AIRPLANE DETECTION ALGORITHM

In our system three different situations in distant airplane detection problem are distinguished depending on the image background. In the first and most trivial case, the aircraft is against clear sky (unstructured background). Naturally this allows the detection in the largest distance. In this case, the aircraft is robustly detectable already when it is larger than 3 pixels. The detection range can be 3.7 km for a Cessna, as reported in [12].

The second case is, when there are clouds behind the aircraft. In this case the more structured the clouds are the more difficult the detection is. In front of dense, quasi homogeneous clouds, aircraft sized 6 pixels can be typically detected. The third case is the terrain background, where aircraft sized larger than 12 pixel can be detected.

Two different kinds of detection algorithms are applied to disjoint regions of the frames depending on the background complexity (Figure 4). The first is for detecting remote aircraft against unstructured background (clear sky or sky with low or medium contrast clouds), while the second was developed for situations, when the background is structured (terrain or high contrast sky background). A local edge density measure classifies the regions of the images and the two

kinds of detections are run on the different regions. The edge density quantifies the amount of edges in a given region of an image. When it is low, the region is considered to be unstructured, otherwise, the region is considered to have structured background. The flowchart of the algorithm is shown in Figure 5.

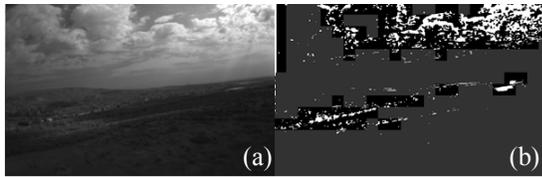


Figure 4. Example frame (a) and its binary edge map (b). On the edge map the darker areas are those where the algorithm for structured background is used. It can be seen that there are areas on the ground which are relatively homogeneous, so it can be defined as unstructured background. And there are clouds in the image which make a part of the sky structured background.

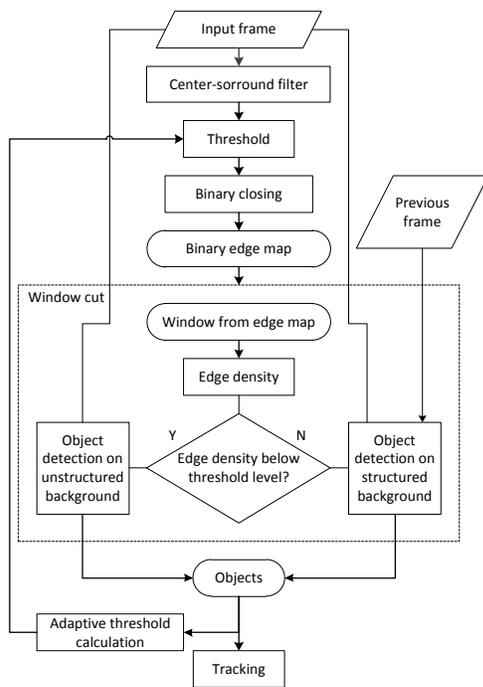


Figure 5. Flowchart of the distant airplane detection algorithm

In order to reduce the computational cost of the algorithm, only the preprocessing is run on the whole frame, after the preprocessing part the more complex algorithms are run on small windows cut from the image. The interesting locations are found based on the binary edge map. The threshold is set in depending on the contrast and the number of found objects. This way the processing system can adapt to various lighting conditions and situations.

In the case of the unstructured background a convolution with a medium sized kernel optimized to extract the small (2-3 pixels large) horizontally elongated blobs is applied. Here the larger than two

detected pixel groups count only, the smaller ones are not considered to be candidate points and discarded. The false candidates are filtered out using local features. The details can be found in [10].

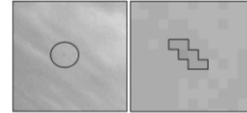


Figure 6. Robustly detectable distant aircraft on a normal sized image and its magnified version. Pixels belong to the aircraft are circumvented. The contrast is 8-12 LSBs.

In the structured background scenario another strategy is needed, as the contrast of the aircraft drops significantly (Figure 7). In this situation, when the object is not detectable on a still image (intra-frame) due to the lack of definite shape, contrast, or color, then its movement should be detected using inter-frame approach.

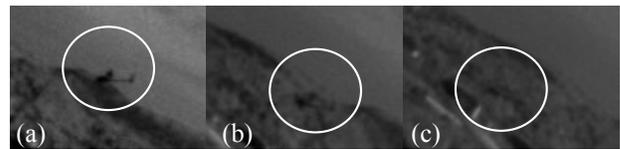


Figure 7. A 20 pixel aircraft is entering terrain background region. It is clearly visible in (a) against sky. Still detectable on (b) against terrain but the contrast is very low. Practically undetectable with intra-frame techniques (c).

Here, due to the moving camera, the standard Gaussian Mixture Modeling cannot be used, however frame differentiation can still be applied. Frame differentiation needs precise background matching when the camera is moving. In our algorithmic framework, due to the limited on-board computational capability we compensate the camera ego-motion with shift only, and not with shift, scaling, and rotation. According to our experiments, the shift only approach still leads to acceptable results, because the objects on the image are in a large distance, therefore they are not enlarging between two frames, and on the other hand, the frame-rate is high enough to keep the effect of rotation very small.

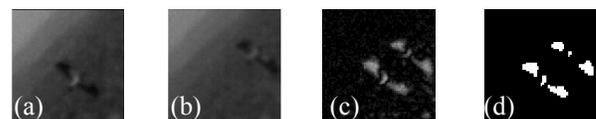


Figure 8. Frame differentiation based moving object detection. (a) and (b) shows the two consecutive matched tiles with a flying bird. (c) shows the absolute difference image. (d) is the thresholded and cleaned image with the twin signature.

The absolute difference image is thresholded first, then with a median filter the small objects are cleared. The remaining white blobs on the thresholded image are

considered as the signatures of the moving objects. As it can be seen in Figure 8, the signature of the moving object appears twice, because one of the twin marks the current position of the moving object, the other marks the previous. Using outputs from previous frames the object can be found on each frame. The last step is the calculation of the center of mass and the size of the signature. These data are used by the tracker.

IV. IMPLEMENTATION ON KILO-PROCESSOR

As it is reported in [10] the algorithm for the detection on unstructured background is implemented on a special FPGA architecture. The main advantages of that system is that it is fast and low power. The main drawback is that the implementation time of a new algorithmic part is high. Thus a GPGPU implementation of the algorithm is started, where the implementation can be done mostly in C/C++ with `opencv::gpu` and only some special function has to be in CUDA language.

Our target system is the nVidia Jetson TK1 development board which consists of the TK1 SoC with the necessary peripherals (SATA, GigE, HDMI, USB, GPIO) and can handle two cameras. This is a low power system with a quad-core ARM Cortex A15 and a Kepler GPU with 192 CUDA cores. The typical power consumption is around 6W which is suitable for a small UAV.



Figure 1. The nVidia Jetson TK1 development board with cameras

The input video stream comes from two cameras ($2 \times 640 \times 480$) and the position and size of intruders are sent through RS232 to the flight control unit. For the image processing the convolutions and morphologic calculations can be accelerated by the Kepler GPU and the IO handling and other algorithmic parts are done by the ARM. An SSD drive is used as a black box for the system. It stores the recorded images and the telemetry data. The SSD drive connects to the system through the SATA interface.

Our first TK1 implementation reaches 6 FPS at 10W (maximized operating frequency), which is suitable for algorithm testing in real situations. Based on the TK1 experiments the improved version of the algorithm will be implemented on FPGA.

V. CONCLUSIONS

In this paper an algorithm for distant aircraft detection is shown. The algorithm uses two strategies running on disjoint part of a frame, one is for the unstructured background and one is for structured background. The frame is partitioned based on the edge density calculated in small windows.

The implementation of the algorithm on the nVidia Jetson TK1 development board is finished and flight test will be run in the near future. This development board is capable of handle two cameras and an SATA SSD. The expected frame rate for the presented algorithm is 6Hz and the power consumption is around 10W including the SSD drive.

ACKNOWLEDGMENT

The support of the ONR Grant (Number: N62909-10-1-7081) is greatly acknowledged. The authors thank Eutecus Inc. for providing the applied tracker algorithm. This research project was also supported by the University of National Excellence Program and the Research Faculty grant awarded to the Pazmany Peter Catholic University, Faculty of Information Technology and Bionics.

REFERENCES

- [1] A. V. Koldaev, "Non-Military UAV Applications," 2007.
- [2] P. Campoy, J. F. Correa, I. Mondragón, C. Martínez, M. Olivares, L. Mejías, and J. Artieda, "Computer Vision Onboard UAVs for Civilian Tasks," *J. Intell. Robot. Syst.*, vol. 54, pp. 105–135, 2008.
- [3] D. Werner, "Making way for unmanned aircraft," *Aerospace America January 2014*, no. January, pp. 28–32, 2014.
- [4] K. Dalamagkidis, K. P. Valavanis, and L. A. Piegler, "On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the National Airspace System," *Prog. Aerosp. Sci.*, vol. 44, no. 7–8, pp. 503–519, Oct. 2008.
- [5] European RPAS Steering Group, "Roadmap for the integration of civil Remotely-Piloted Aircraft Systems into the European Aviation System," Rep., 2013.
- [6] International Civil Aviation Organization, "Procedures for Air Navigation Services: Air Traffic Management," Doc 4444 ATM/501, Nov. 2007.
- [7] International Civil Aviation Organization, "Airborne Collision Avoidance System (ACAS) Manual," Doc 9863 AN/461, 2012.
- [8] D. Accardo, G. Fasano, L. Forlenza, A. Moccia, and A. Rispoli, "Flight Test of a Radar-Based Tracking System for UAS Sense and Avoid," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 1139–1160, Apr. 2013.
- [9] J. Lai, J. J. Ford, P. O'Shea, and L. Mejías, "Vision-Based Estimation of Airborne Target Pseudobearing Rate using Hidden Markov Model Filters," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2129–2145, Oct. 2013.
- [10] A. Zarandy, Z. Nagy, B. Vanek, and T. Zsedrovits, "A five-camera vision system for UAV visual attitude calculation and collision warning," *Comput. Vis. Syst. Lect. Notes Comput. Sci.*, vol. 7963, pp. 11–20, 2013.
- [11] B. Vanek, T. Péni, Á. Zarándy, J. Bokor, T. Zsedrovits, and T. Roska, "Performance Characteristics of a Complete Vision Only Sense and Avoid System," in *AIAA Guidance, Navigation, and Control Conference*, 2012.
- [12] T. Zsedrovits, A. Zarandy, B. Vanek, T. Peni, J. Bokor, and T. Roska, "Visual Detection and Implementation Aspects of a UAV See and Avoid System," in *2011 20th European Conference on Circuit Theory and Design (ECCTD)*, 2011, pp. 472–475.