Embedded vision system for automated drone landing site detection – a demo

Patryk Fraczek
AGH University of Science
and Technology Krakow, Poland
E-mail: vakii@student.agh.edu.pl

Andre Mora
CTS/UNINOVA, FCT, University
NOVA of Lisbon, Caparica, Portugal
E-mail: atm@uninova.pt

Tomasz Kryjak, *Member IEEE*
AGH University of Science
and Technology Krakow, Poland
E-mail: tomasz.kryjak@agh.edu.pl

***Abstract*—This paper presents an embedded video system able to detect safe landing sites for the purpose of a drone automated landing procedure. The solution was implemented on a heterogeneous Zynq SoC (System on Chip) device from Xilinx and a Jetson GPU (Graphic Processing Unit + ARM processor system). Site detection and terrain classification (using decision trees), as well as shadow detection is based on colour and texture features. The proposed system is able to process 1280 x 720 @ 60 fps video stream in real-time on Zynq SoC.**

***Index Terms*—Unmanned Aerial Vehicle (UAV), safe landing site detection, decision trees (DT), machine learning, digital image processing, FPGA, Zynq, GPU**

I. Introduction

The design and implementation of a fully autonomous drone is a challenging task. One of its important aspects is an automated landing procedure. Not only a safe landing must be ensured in the case of normal conditions, but what is more important (considering reliability) in the event of emergency situations (components failure, low battery, strong wind or other poor weather conditions, etc.). According to [1] and [2] robust terrain classification is an essential element of this kind of applications. Suitable are only concrete, asphalt or ground areas. Secondly, they should be flat without any significant slopes and big enough for the drone to land. Finally, the presence of shadows should be taken into account, as well as the level of battery charge (reachability). In general the closest suitable locations should be preferred. This article presents the result of research on implementing such a system on a Zynq SoC device and embedded GPU.

II. The proposed algorithm

Choosing a safe landing site is a multi-criteria decision problem. Water should be avoided, as well as, moving objects, trees and rooftops. It is assumed that the algorithm should use just an on-board camera (a conventional RGB image sensor) to derive landing suitability maps and then choose the safest landing site.

The scheme of proposed algorithm is shown in Figure 1. The solution is based on results presented in [1], [2] and [3]. Four modules can be distinguished: pixel classification, shadow detection, reachability map and data fusion.

The work presented in this paper was supported by the National Science Centre project no. 2016/23/D/ST6/01389 and Fundação para a Ciência e a Tecnologia under the grant SFRH/BSAB/135037/2017

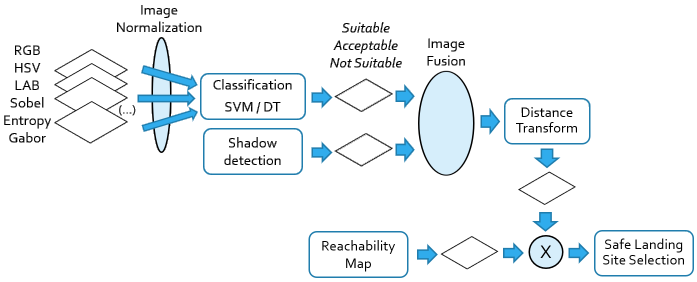

Fig. 1. The proposed drone landing site selection system architecture

In the solution, a machine learning approach was used to classify the terrain type. The source of video data were sequences recorded by drone on-board cameras available on YouTube. Then 21×21 pixels patches of different terrain samples were manually selected. In total, 3079 training samples were used. Next, colour and texture feature extraction was applied. Contexts of 7×7, 9×9 and 11×11 were analysed. Several different common methods were used: entropy, Gabor filters, Sobel, GLCM (grey-level co-occurrence matrix). Three classes of landing suitability were distinguished: class 2 – best landing sites (soil, asphalt, concrete), class 1 – alternative landing sites (grass), class 0 – not suitable for landing (trees, water, etc.) Two classifiers Support Vector Machines (SVM) and Decision Trees (DT) were considered. Based on the preliminary training results, the following feature set was determined: **Pixel Colour** – mean, minimum and maximum from colour spaces (RGB, YCbCr, HSV, CIE Lab) except L component, **Pixel Texture** – mean from Sobel 3x3 neighborhood. The number of features considered was eventually 34 comparing to 117 initial ones.

Finally, the 9×9 neighbourhood was selected and data normalized – all features were represented by 8-bit number. The cross-validation evaluation revealed that DT perform better (92.1%) compared to SVM (83.7%).

Another part of the solution was shadow detection based on global binarization with a threshold determined on the basis of histogram analysis (first local minimum after first local maximum). Also the reachability layer intends to prefer locations closer to the drone – due to chosen safety requirements. A 2D Gaussian distribution was used for this purpose.

In another stage, the output map data from individual modules was fused. The basis was the I_c map with the pixel classification results. For each point in I_c for which shadow detection occurred in I_{sh}, there was a reduction of suitability class by 1 (i.e. from 2 to 1 or from 1 to 0). After examination of several approaches for finding suitable landing areas a distance transform based one was used. It allows to create a map, without plateaus, where each pixel is a function of its value and the distance to the closest plateau border. This property allows to find circular areas of selected radius. In the current version this threshold is fixed, however if a 1D LIDAR device would be used, this value should depend on the current altitude over ground.

Finally, the distance transform map is analysed and locations exceeding the threshold are stored in a priority queue sorted against the distance to the image centre (i.e. to the drone). The first location in the queue is considered as the landing place. It should be noted that the above described procedure is applied in two stages. First, only class 2 locations are considered. Then, if no landing place is detected, classes 2 and 1 are fused (reducing class 2 suitability to 1) and the distance transform computed again.

III. Hardware-software implementation

The described system was verified on two hardware platforms: heterogeneous Zynq SoC (Arty-Z7 20) and on the embedded GPU platform NVIDIA Jetson TX2. Both solutions were designed to process a 1280×720 video stream. Programming and configuration of the Arty-Z7 was conducted using the Xilinx Vivado 2017.4 and the software for GPU was implemented in C++ and profiled using NVIDIA Nsight.

In the first implementation step on Zynq SoC, the algorithm was split into a hardware and software part. The proposed architecture is presented in Figure 2 The hardware part contains simple contextual calculations and paralleled task like finding minimum or maximum. Also Look-up Tables were used as a substitute for complicated calculations (converting from RGB to Lab colour space). The software part (run on the general purpose processor) contains elements of algorithm which are sequential or iterative (distance transform).

Resource utilization for the FPGA part reported be the Vivado tool: LUT: 33270/53200 (62.54 %), FF: 58569/106400 (55.05 %), BRAM: 122/140 (86.79 %), DSP: 22/220 (10.00 %) and the power usage estimated is about 3.5 W (FPGA + ARM processor). For GPU the power consumption is 7.5 W (under typical load).

In the current version the system uses about 60% of the available resources. In the FPGA part all operations are performed in real-time – 60 frames per second. However, the distance transform and landing site detection on the ARM core requires 200-400 ms for a single execution. On the other hand, the mean processing time of a single frame for the GPU was 1.2 s. One reason for the quite low performance were two custom CUDA functions (not from OpenCV). However, even without them only 8 frames per second could be processed.

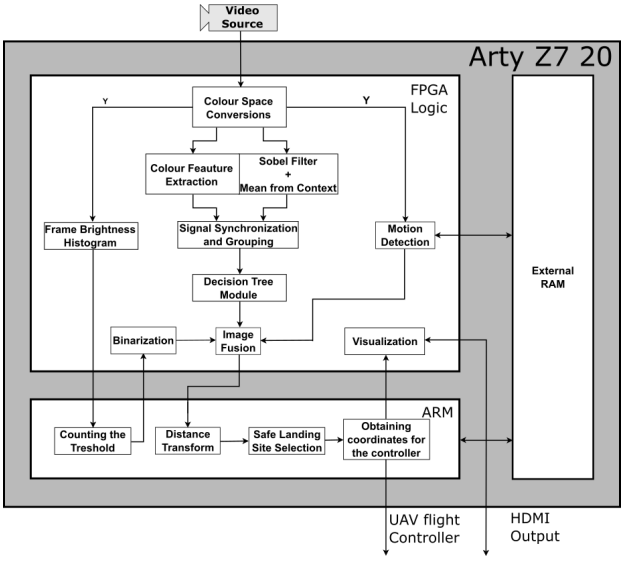

Fig. 2. Scheme of the proposed hardware-software implementation of the system in Zynq SoC

From a practical point of view this is not a big problem, as tracking the landing site location just few times per second is acceptable. On the other hand, implementing the distance transform in hardware should allow to obtain real-time processing, even for higher image resolutions.

IV. Summary

In this paper an algorithm for the detection of landing sites for an autonomous drone was presented. The proposed solution has been verified on 100 test images and in over 80% cases provided a correct result. The system has been implemented on two embedded platforms: Zynq SoC and Jetson TX2. The performance comparison showed that for this application the Zynq SoC is a better solution, as it offers greater computing performance with less energy usage. As part of the further development of the system, it is planned to add a moving object segmentation (with drone ergo-motion compensation) and module extracting 3D information. Implementations on embedded platforms could be further optimized (e.g. distance transform). Another interesting matter to do would be to replace DT with embedded CNNs - they could allow more precise classification (more than 3 labels). Finally, a challenge would be to run the system for a 4K image in real-time.

References

[1] Fitzgerald, D., et al.: A Vision Based Emergency Forced Landing System for an Autonomous UAV. Proceedings Australian International Aerospace Congress Conference, Melbourne, Australia. (2005)

[2] Mukadam, K., et al.: Detection of landing areas for unmanned aerial vehicles. In 2016 International Conference on Computing Communication Control and automation (ICCUBEA). IEEE, pp. 15. (2016).

[3] Mora, A., et al.: Land Cover Classification from Multispectral Data Using Computational Intelligence Tools: A Comparative Study. Information 8 (4): 147. (2017)

[4] Saqib, F., et al.: Pipelined Decision Tree Classification Accelerator Implementation in FPGA (DT-CAIF), IEEE Transactions on Computers, vol. 64, no. 1, pp. 280-285, (2015)