CEAS EuroGNC 2019
"Conference on Guidance, Navigation and Control"
3-5 April 2019 @ Politecnico di Milano, Milan, Italy.

CEAS-GNC-2019-072

Multi-Modal Image Processing Pipeline for a Reliable Emergency Landing Field Identification

Andreas Klos, Jörg Lenhardt, Marius Klein, Wolfram Schiffmann

Abstract If the pilot of an aircraft is forced to perform an emergency landing, quick and reliable decisions regarding the flight path are necessary. Besides, it is not guaranteed that a published landing field is located within reach. In such a situation the selection of an appropriate emergency landing field denotes a crucial task for the pilot. The choice of a suitable emergency landing field influences the damage of the aircraft, the civil population, the crew as well as passengers on board. Based on public available geodata, we developed an application that automatically identifies emergency landing fields by an appropriate operation sequence of an image processing pipeline. Our approach is based on satellite imagery, rasterized road maps, and interpolated digital elevation models. The chosen image processing pipeline consists of eight consecutive steps. The results proved that our approach is capable of a reliable identification of appropriate emergency landing fields for a certain parametrization of the applied algorithms. The found emergency landing fields are stored in a MySQL database for fast access, even in the case of instrument meteorological conditions.

Key words: image processing, emergency landing field, aviation, elevation data processing, satellite imagery processing

Andreas Klos
Faculty of Mathematics and Computer Science, FernUniversität in Hagen, Germany, e-mail: `Andreas.Klos@fernuni-hagen.de`

Jörg Lenhardt
Faculty of Mathematics and Computer Science, FernUniversität in Hagen, Germany, e-mail: `Joerg.Lenhardt@fernuni-hagen.de`

Marius Klein
Faculty of Mathematics and Computer Science, FernUniversität in Hagen, Germany, e-mail: `Marius.Klein@fernuni-hagen.de`

Wolfram Schiffmann
Faculty of Mathematics and Computer Science, FernUniversität in Hagen, Germany, e-mail: `Wolfram.Schiffmann@fernuni-hagen.de`

1 Introduction

A total engine failure during a manned flight poses a major threat to the passengers, air crew and aircraft. In such cases, the pilot is forced to perform an emergency landing (EL). If the pilot fails during such an extreme situation, the manoeuvre may even result in a lethal accident.

The Federal Aviation Administration reported that the number of fatal accidents in General Aviation (GA) has decreased in recent years. Nevertheless, 347 people died in 209 GA accidents in 2017. Hence, a significant reduction of lethal GA crashes depicts a crucial issue. The decrease of fatal flight accidents has been extensively studied and many key techniques were developed to support the pilot in its decision process. These ranges from various path planning approaches – e. g. in [1] – to the automatic detection of Emergency Landing Fields (ELF).

Fortunately, if the pilot is compelled to perform an unplanned landing, e. g. caused by the total loss of thrust, the aircraft is still capable to perform a glide. The maximum glide range is determined by the kinetic energy (velocity) and potential energy (altitude). Even for highly trained pilots it is challenging to estimate appropriate ELFs.

The choice of a suitable ELF is be fundamental for the survival of the people on-board and determines the resulting damage of the aircraft. For this purpose, the pilot's decision about the ELF depends basically on: Size, Shape, Slope, Surface, Surrounding and Civilization [2].

Hence, the main objective of our approach is to simplify and accelerate the decisions of the pilot during an emergency situation regarding the choice of a proper ELF. To achieve our goal, we process multi-modal geo-data with various well known and in-house developed image processing techniques. Thereby, each pixel in the considered area is evaluated referred to its suitability for an EL. After the selection of landable areas the size and shape of the runways are examined by considering the minimum length and width for a certain aircraft. If the considered landable area contains pixels which are already classified as landable, the ELF is further investigated. Afterwards, the surface conditions are examined with respect to the slope. Subsequently, the candidate ELF area has to be validated concerning to the obstacle clearance by taking the gliding angle into account.

The paper is organized as follows: Section 2 is dedicated to the related work. Section 3 describes the applied machine vision algorithms. Section 4 presents the data model. In Sec. 5 the achieved results are presented and discussed. Finally, the proposed findings are concluded and an outlook on our future works is given Sec. 6

2 Related Work

For all kinds of aircrafts the selection of an appropriate ELF during an EL constitutes a crucial task for the pilot. Therefore, several supporting image acquisition and processing algorithms have been developed in recent years.

The image processing techniques can be subdivided in three categories: 1) processing real-time images obtained by on-board sensors; 2) processing pre-acquired data; 3) processing multi-modal image sources.

In [3] a two-stage segmentation approach based on satellite imagery is proposed. First, an initial segmentation is performed by analyzing the corresponding histogram of the considered satellite imagery to estimate the number of different classes. Afterwards, a structure preserving segmentation is performed in the spectral domain. Nevertheless, this approach lacks in its ability of edge detection and disregards the suitability for an EL in its dimensions. In [4] another two step processing algorithm is presented which performs first a sectioning of the considered region – Canny edge, line growing – and afterwards a geometric check to ensure the suitability regarding to the shape as well as dimension of the examined region. Unfortunately, these image processing steps omit to analyze the slope and bumpiness which are required to guarantee the suitability of located ELFs.

In [5] a digital elevation model processing approach is introduced which performs the examination by a quadtree data structure. Thereby, the data is separated till a minimum dimension of the runway is reached. Subsequently, the average altitude and the variance within each leaf is calculated. Unfortunately, the metric of variance and average altitude may be insufficient for the selection of a suitable ELF because outlier – caused e. g. by buildings – might be pruned which could lead to a false-positive classification of the corresponding region. Besides, the lack of investigation of the ELF's surface could hide e. g. water areas which depicts also a weakness to the algorithm proposed in [6] where only elevation data is processed regarding to predefined slope restrictions.

For that reason, the classification of the surface of the ELFs is considered as crucial as presented in [7]. Thereby, the classification is performed by the application of a multi-class Support Vector Machine (SVM). Other terrain classifications are proposed in [8] – based on SVM and AdaBoost for multi-spectral images, in [9] – rest on SVM and multi layer perceptron, and in [10] – premised on SVM and Random Forrests processing monocular camera data. In [11] a surface classification algorithm is introduced which applies standard image processing techniques and artificial neural networks to verify obstacle clearness.

In [12] an *k*-Nearest Neighbor approach is proposed that considers a feature vector generated by standard image processing – e. g. histogram thresholding, Canny edge etc. – of data acquired by an UAV camera and light intensities measured by a light intensity sensor. Nevertheless, only small areas within the field of view are analyzed and only a highly restricted number of suitable ELFs can be found. Furthermore, this approach depends on the weather conditions. These drawbacks are revised in [13] by aircraft-mounted cameras oriented to the front and a horizon detection algorithm to identify the ground in the images. Besides, they apply a nonlinear retinex image-enhancement method to revamp the environmental effects and to improve the contrast and sharpness. However, the results heavily dependent on the resolution of the aircraft-mounted cameras and the altitude of the aircraft.

Hence, a combination of processing images from an aircraft-mounted camera and pre-acquired digital elevation models is shown in [14]. The authors investigated the processing of 2D geodata and reconstructed 3D model. In [2] another multi-modal processing algorithm is proposed. First, preliminary processing steps are performed as mentioned earlier in [4]. Afterwards, man-made and natural objects were distinguished by considering the intensity values. Subsequently, the geometric shape, the surface type and the slope are considered. Unfortunately, the obstacle clearance of the final approach is left unconsidered.

Our contribution in the considered research area consists of a new Multi-Modal Image Processing Pipeline (MMIPP), the verification of obstacle clearance of the final approach and the provision of a database which contains the achieved results.

3 Machine Vision Algorithms

The proposed Machine Vision (MV) algorithm is based on three different data sources: elevation data, satellite imagery, and road map. These data was made available by the Google Static Maps Application Programming Interface (API) and the Google Maps Elevation API. The data sources are denoted by I in the further reading. First, we assume that every data pixel of the considered area is landable. Of course, this assertion is incorrect for huge parts of the area. Therefore, the area which is sufficient for an EL has to be restricted. This is achieved by a pipelined application of multiple MV operations. Every MV method operates on a predetermined data layer. If an area is marked as inappropriate for an EL, this is valid for all subsequent MV operations.

3.1 Color Exclusion Method, Segmentation, and Edge Detection

Color Exclusion Method: Every pixel is evaluated regarding a predefined color threshold. If the pixel exhibits the threshold, it is marked as insufficient for an ELF. This method is performed in the RGB and HSV color space. In RGB color space the thresholds for red, green, and blue color – within the considered pixel is inappropriate for an ELF – has to be determined. This approach is used e. g. to exclude water aerials from the examined area. In HSV color space the thresholds for hue, saturation, and value – within the investigated pixel is unsuitable for an ELF – has to be parametrized. This approach is used e. g. to exclude man made areas.
Segmentation: After each processing step, a segmentation of the processed data layer is performed. Initially, every pixel refers to the same segment as long as a previous determined condition of the neighboring pixel elements is fulfilled. This algorithm is applied on roadmaps which are differently colored for the various kinds of areas, e. g. blue denotes water, white tags streets etc. After applying the previous described color difference method with a threshold of zero for the three color channels, the roadmap is partitioned into many independent segment. The segmentation is followed by a counting operation of the number of pixels in each segment. If the

number of pixels is smaller than the minimum number of pixels required by the minimum dimension of the ELF, the considered segment is excluded from the subsequent image processing steps and is tagged as invalid for an ELF. The counting operation of the pixels inside a certain segments is performed in $O(n)$.

Edge detection: Homogeneous areas in satellite imagery or roadmaps are commonly suitable for an EL. Areas characterized by sudden intensity changes – which can be identified by an edge detection algorithm – are often unsuitable for an EL. First, the chosen layer is converted to a gray scale image followed by the application of Canny edge detection. The edge detection by Canny was initially introduced in [15]. Thereby, first the gray scale image is smoothed by the convolution with a Gaussian filter of size 5×5 and the standard deviation of 1.4 Subsequently, the magnitudes and directions of the edges are computed.The gradient direction is always perpendicular to the direction of the edge and is rounded to one of four angles representing the horizontal, vertical, and two diagonal directions. Afterwards, a non-maximum suppression is performed to remove unwanted pixels which do not belong to an edge. Therefore, every pixel is evaluated, if a local maximum is in the neighborhood in gradient direction. Finally, the hysteresis step decides which pixel corresponds to an edge and is considered as a strong edge. For that reason, a lower and higher threshold is required.

3.2 Variance Exclusion and Local Difference Methods

Variance exclusion method: The variance of pre-segmented image areas is computed and the data points are classified as landable if the variance is below the threshold. First, the mean and subsequently the variance of each segment is calculated. In our work, the variance exclusion method is based on the normal vectors between the elevation value of each pixel. The normal vector $\mathbf{n}_i$ with $i \in \{1,2\}$ is calculated by the cross product of the vectors $\mathbf{v}_i$ with $i \in \{1,2,3,4\}$ which are spanned from the considered pixel position to the pixels in the neighborhood as shown in Fig. 1 where $\mathbf{v}_1 - \mathbf{v}_4$ are the spanned vectors from the considered pixel to the neighboring pixels. $\mathbf{n}_1$ and $\mathbf{n}_2$ are the determined normal vectors. By doing so, eight normal vectors are calculated. After that, the normal vectors are summed up and normalized

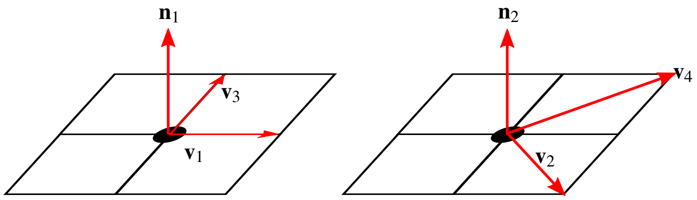


Fig. 1 a) $\mathbf{n}_1$ calculated as $\mathbf{v}_1 \times \mathbf{v}_3$; b) $\mathbf{n}_2$ computed as $\mathbf{v}_2 \times \mathbf{v}_4$.

to a new vector $\hat{n}_i^s$ with $i \in \{1,...,m\}$ and s as index for the considered segment. This calculation is done for every pixel in the considered area segment. Subsequently, the mean value of the normalized vectors $\bar{n}$ for that area is determined as shown in Eq. 1

$$\bar{n} = \frac{1}{m}\sum_{i=1}^{m}\hat{n}_i^s \tag{1}$$

where m denotes the number of pixels in the investigated area segment. Afterwards, the angle α_i between $\bar{n}$ and $\hat{n}_i^s$ is computed as presented in Eq. 2

$$\alpha_i = \arccos(\bar{n}\cdot\hat{n}_i^s)\cdot\frac{180^\circ}{\pi} \tag{2}$$

Finally, the variance is calculated. Therefore, variance equation becomes adopted as shown in Eq. 3

$$s^2 = \frac{1}{m-1}\sum_{i=1}^{m}\alpha_i^2 \tag{3}$$

where α_i is determined as show in Eq. 2 and m denotes the number of pixels in the observed area segment. This algorithm enables a decision about the roughness of each inspected segment area. If these variance s^2 crosses a certain threshold, the area is excluded in any further processing step and will be marked as inconvenient as ELF.

Local difference methods: Basically, this method analyzes the relationships between neighboring pixel elements and can be applied on all three previous mentioned data layers. Generally, a point of the data grid – more details in Sec. 4 – is chosen and pushed on a stack. The elements of the stack are processed by last in, first out principle. Consequently, the last element which was pushed on the stack becomes analyzed first and is considered as central element. First, it has to be ensured that the neighboring elements – elements on the stack except the last one – untreated, still marked as suitable for an ELF and unsegmented. If this assumptions are valid the neighbor will be assigned to the same area segment as the central element and pushed on a stack. If the neighborhood requirements are unsatisfied, the neighboring element will be marked as inappropriate for an ELF. This procedure is repeated till the stack is empty. If the stack is empty, but elements remain on the data grid which are still untreated, the described process is repeated for these elements with the assignment to a different area segment. The various local difference methods are depicted in the following.

The **slope difference method** is applied on the elevation data layer. As metric for the neighborhood, the angle between connecting line of two adjacent elevation elements and the horizontal line, as illustrated in Fig. 2 is used where α is the metric for the slope between two adjacent elevation elements. h_1 and h_2 are the elevation values and d_{12} denotes the distance between the two neighboring elevation elements. Thereby, the angle α is calculated as shown in Eq. 4

$$\alpha = \arctan\left(\frac{|h_1 - h_2|}{d_{12}}\right)\cdot\frac{180^\circ}{\pi} \tag{4}$$

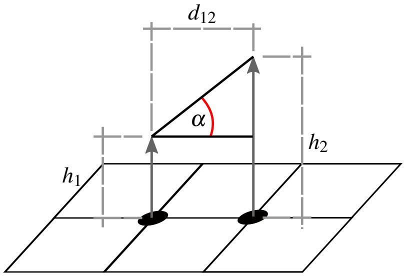


Fig. 2 α is the slope between two adjacent elevation elements; h_1 and h_2 are the elevation values at the corresponding pixel positions; d_{12} is the distance between the two neighboring elevation elements.

The **color difference method** processes color information of the current input data and is applied to the satellite images and roadmaps. The color difference method is implemented for RGB and HSV color space. For application in RGB color space, thresholds for the red, green, and blue has to be chosen independently. If none of the thresholds is exceeded, the neighborhood relation is classified as valid. The same applies for the application in the HSV color space expect that thresholds for hue, saturation and value has to be defined. The evaluation of the result is treated like explained earlier.

The **angle difference normal vector method** is suitable for processing elevation data. Thereby, sharp slopes are detected and marked as unsuitable for an ELF. This is facilitated by computing the normal vectors for every data element, as explained in Sec. 3.2 The angle between to normal vectors is used as neighborhood relation requirement. To take the map projection into account, the angle is weighted by the distance between two data elements. If the weighted angle exceeds a certain threshold, the neighborhood relation is invalid for an ELF.

3.3 Emergency Landing Field Identification

First, the **search of potential ELFs** is performed. This requires the following parameters: Minimum dimension of the ELF; distance between two consecutive reference points (RP) – RP denotes the center of a ELF; sampling interval of the ELF direction; max. slope and descent threshold of the considered ELF. Initially, the RPs are placed in each segmented area with respect to the space between neighboring RPs. Afterwards, the min. size of the runway is considered by placing a rectangle around the RP regarding search direction (sd). If the minimum required dimension is fulfilled and only pixel classified as landable are contained, the total slope and descent along as well as transversal to the ELF direction are calculated. Provided that the slope and decent is smaller than the pre-defined threshold, the coordinates of the RP, start and end point as well as the considered sd are temporarily saved. Subsequently, an evaluation and optimization of the ELFs is performed.

Secondly, the **evaluation and optimization of the temporary ELFs** is executed. This requires the definition of runway's final approach sector opening angle, the test distance (starting from the ELF threshold) and the aircraft's glide angle. An evaluation of each ELF is performed by validating the obstacle clearance of the final approach sector within the determined test distance. The evaluation of the obstacle clearance is performed on the elevation data. If the glide path – estimated by the glide angle – intersects the surface of the elevation data, the investigated ELF is assumed as unsuitable for an ELF. Otherwise, the ELF becomes optimized in the subsequent step. Thereby, the dimension along and against the direction of the ELF is enlarged. After each increase of the ELFs dimension, the ELF is validated regarding to the maximum difference between the new start and end point of the runway along the center of the ELF in longitudinal direction and the obstacle clearance. This process is repeated n-times until a threshold is exceeded or the obstacle clearance of the opening angle is violated. In this case, the new start and end point coordinates of optimization step $n-1$ are stored in a MySQL-Database as well as direction and coordinates of the RP.

4 Data Model

The developed Data Model (DM) consists of multiple layers as shown in Fig. 3 Therein, the DM is shown with three different layers. The uppermost consists of satellite imagery, the middle one is composed of the corresponding roadmap and the bottom layer contains the dedicated elevation data. The data in our model is requested by the usage of the APIs mentioned in Sec. 3. Thereby, the corresponding cylindrical map projection is taken into account to achieve a correct mapping of the requested geographical data. Therefore, Eq. 5 must be solved for the longitude λ

$$x = \frac{\lambda + 180^\circ}{360^\circ} \cdot 256 \cdot 2^z \tag{5}$$

where x is the Cartesian x-component of the considered pixel position and z denotes the chosen zoom factor as described in [16]. As a result, Eq. 6 is obtained.

$$\lambda = \frac{x \cdot 360^\circ}{256 \cdot 2^z} - 180^\circ \tag{6}$$

Afterwards, Eq. 7 must be solved for the latitude ϕ

$$y = \left(0.5 - \frac{\log_{10} \frac{1+\sin\left(\phi \cdot \frac{\pi}{180^\circ}\right)}{1-\sin\left(\phi \cdot \frac{\pi}{180^\circ}\right)}}{4\pi} \right) \cdot 256 \cdot 2^z \tag{7}$$

where y is the Cartesian y-component of the analyzed pixel position. z denotes the chosen zoom factor as mentioned earlier. The result is shown in Eq. 8

$$\phi = \arcsin\left(1 - \frac{2}{10^{\left(0.5 - \frac{y}{256 \cdot 2^z}\right) \cdot 4\pi} + 1}\right) \cdot \frac{180^\circ}{\pi} \tag{8}$$

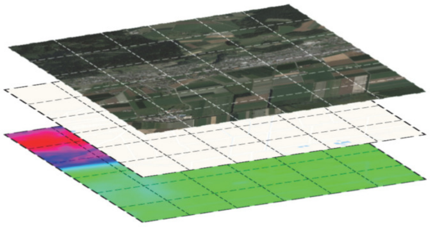


Fig. 3 Data model composed of three layer: Lowest is the elevation layer; Middle is the roadmap layer and the uppermost is the satellite imagery layer.

Furthermore, the resolution of the elevation data is lower compared to the other layers. In order to achieve an assignment of an elevation value to every pixel of the imagery a bilinear interpolation is performed between four elevation data points: $B_{j,k}, B_{j+1,k}, B_{j,k+1}, B_{j+1,k+1}$. This technique is composed of the interpolation in x- and y-direction. The first interpolation in the horizontal direction is shown in Eq. 9

$$\begin{aligned} B_{j+x,k} &= (1-x) \cdot B_{j,k} + x \cdot B_{j+1,k} \\ B_{j+x,k+1} &= (1-x) \cdot B_{j,k+1} + x \cdot B_{j+1,k+1} \end{aligned} \tag{9}$$

where x denotes the shift amount within the four elevation points. The second interpolation in the vertical direction is given by Eq. 10

$$B_{j+x,k+y} = (1-y) \cdot B_{j+x,k} + y \cdot B_{j+x,k+1} \tag{10}$$

where y is the shift amount within the four elevation points in vertical direction. For further reading [17] is suggested. After the creation of the DM each layer is processed with a certain MV-technique as described in Sec. 3. The MMIPP and the results are presented and discussed in Sec. 5.

5 Results and Discussion

Our objective is the identification of ELFs and their provision in a database. Thus, even in instrument meteorological conditions the decisions of the pilot can be supported in an emergency situation. First, the satellite imagery, roadmaps, and digital elevation model are queried by the usage of APIs provided by Google. Afterwards, the DM introduced in Sec. 4 is created.

Thereafter, the considered aircraft model is chosen to determine the minimum requirements regarding the dimensions of the ELF. As a sample aircraft a Diamond DA20-C1 is selected. In [18, p. 5 - 17] the minimum required length of an ELF is stated as 214 m by assuming international standard atmosphere, maximum take off mass and 610 m height as mean sea level. This specification is invalid for ELFs covered by grassland, fields or other surfaces. As a consequence, the recommended length for the chosen aircraft model in the corresponding manual has to be adjusted. The manual omit how the runways length should be adjusted. The manual of the Diamond DA40 D recommends to increase the length of the ELF about 25% in cases of unpaved ELF with grass longer than 10 cm which results in a new length of 267.5 m. Furthermore, the manual describes to increase the length of the ELF about 10% for an inclined ELFs of about 2% which corresponds to a gliding angle of 1.1° and results in a new minimum length of 294.25 m [19, p. 5 - 19]. The minimum width of the ELF was determined as the triple of the aircrafts span (32.67 m). Subsequently, the best configuration of our MMIPP is investigated.

The best configuration is determined by executing all permutations of the MMIPP. Thereby, the potential ELF identification, the ELF evaluation and optimization is fixed in the processing pipeline as highlighted in bold as step seven and eight in Tab 1. Thus, $6! = 720$ possible permutations remain. To restrict the computational demand the local color difference and color exclusion technique remain fixed in there position of the MMIPP. This is emphasized in bold for step one and two in Tab. 1 Thus, $4! = 24$ permutations are left for further investigations. The choice of the best configuration of the MMIPP is based on the following two metrics: Execution time and number of identified ELFs. Thereby, the main objective is minimum execution time and lowest count of identified ELFs. As a consequence of the minimum number of identified ELFs it is assumed that the probability of a false/positive classification – expect to be the worst case scenario regarding to ELF identification – is lesser as in other permutations. The determined best configuration is shown in Tab. 1 which contains the assignment of each processing step to a certain data layer and color space. Furthermore, the execution time of the individual processing steps and the whole processing time for the investigated five km^2 area at location (N: 51.54056°, E: 7.94889°) are presented and summed up to 1’193.581 s. It is obvious that the search of potential ELFs with 474.03 s and their evaluation and optimization with 442.647 s take the major amount of time during the execution of the MMIPP. Table 2 presents three permutations of the proposed MV-techniques where the first permutation depicts the configuration as introduced in Tab. 1 The remaining two permutations are the shuffled order of the processing steps regarding the best configuration shown before. The sequence variations of the MMIPP are highlighted in gray color. The second permutation represents

Table 1 Best configuration of the developed MMIPP with corresponding execution times.

Step	Algorithm	Layer	Color space	Exec. time [s]
1	**Local color difference**	**Roadmap**	**RGB**	**62.704**
2	**Color exclusion**	**Roadmap**	**HSV**	**52.519**
3	Edge exclusion	Satellite imagery	RGB	68.468
4	Local color difference	Satellite imagery	HSV	43.943
5	Variance exclusion	DEM	–	121.659
6	Local slope difference	DEM	–	42.834
7	**Potential ELF identification**	**Segmented map**	**RGB**	**474.03**
8	**ELF evaluation and optimization**	**Segmented map**	**RGB**	**442.647**
Total execution time:				1’193.581

Table 2 Summarized execution times (ET), excluded area (EA), number of valid reference points (Nr. RP) and number of optimized ELFs (Nr. ELF) for three permutations of the MMIPP.

Permutation 1			Permutation 2			Permutation 3		
PT	EA [%]	ET [s]	PT	EA [%]	ET [s]	PT	EA [%]	ET [s]
Step 1	8.03	62.704	Step 1	8.03	62.704	Step 1	8.03	62.704
Step 2	8.03	52.519	Step 2	8.03	52.519	Step 2	8.03	52.519
Step 3	22.67	68.468	Step 6	9.97	57.018	Step 3	22.69	68.453
Step 4	28.19	43.943	Step 5	15.2	150.574	Step 5	26.18	129.032
Step 5	32.91	121.659	Step 4	22.92	48.551	Step 6	27.54	46.286
Step 6	34.12	42.834	Step 3	31.48	62.532	Step 4	32.24	41.537
Step 7	—	474.030	Step 7	—	560.151	Step 7	—	653.613
Step 8	—	442.647	Step 8	—	755.933	Step 8	—	1035.024
∑ET [s]	1193.581		∑ET [s]	1634.759		∑ET [s]	1973.945	
Nr. RP	4		Nr. RP	5		Nr. RP	5	
Nr. ELF	9		Nr. ELF	11		Nr. ELF	11	

the order of the processing methods where the smallest amount (about 31.48%) of the considered area was denoted as unsuitable for an EL. Obviously the steps 3 to 6 are in reversed order relating to permutation 1. The processing differences of the results obtained by the first and second permutation are illustrated in Fig. 4 where the final segmentation result of step 1 - 6 is subtracted from the segmentation output of the best configuration which was the permutation with the greatest amount of area classified as inappropriate for an EL (34.12%). The red color denotes areas classified as suitable for an EL by the first and excluded by the second permutation. The green color shows the contrary, these are areas which are tagged as landable by the second and excluded by the first permutation. It can be seen, that more areas are classified as convenient for an EL by the second than by the first permutation. Hence, permutation 2 placed one more RP and identified two further ELFs. Permutation 3 represented in Tab. 2 has the longest total execution time (∑ET). The ∑ET of the best and third permutation differ in 780.364 s which is 65.38% of the first permutation's processing time. Step 5 takes the longest duration among processing steps 1 – 6. Considering the amount of EA after those steps – lower in permutation 2 and 3 compared to the first one – it is obvious that the processing time of step 7 is closely related to the area excluded by the previous steps. The ET of step 8 depends on the number of identified potential ELFs. With application of permutation 2 and 3 eleven valid ELFs are identified whereas the execution of permutation 1 is able to find nine ELFs suitable for an EL.

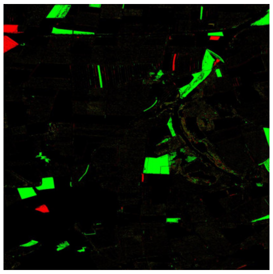


Fig. 4 Red: permutation 1 classified as landable, permutation 2 classified as excluded; green: contrary case.

Previous results lead to the conclusion that the selection of the MMIPP's best configuration is a multi-objective optimization problem. While being anxious to reduce the ∑ET, the number of detected ELFs should be small to avoid false/positive classifications.

The results of the proposed MMIPP are shown in Fig. 5, where eight partial figures are shown and labeled from (a) to (h). In Fig. 5 (a) the processing result of the local color difference method the initial segmentation operating on the roadmap data in RGB color space is shown. Red regions mask areas which are potentially landable and parts of the image classified as unsuitable for an EL are colored differently (this color notation is valid for Fig. 5 (a) - (d)). The algorithm is parametrized as follows: R = 0, G = 0, B = 0. If one of the thresholds is exceeded, the pixel is assigned to another class. Afterwards, each segment is evaluated regarding the number of included pixels. Obviously, there are certain regions in the area containing a smaller number of pixels than required by the minimum dimensions of the ELF.

The result of step two is presented in Fig. 5 (b). The main objective is the removal of water areas. Due to the lacking water in the examined area, the result is exactly the same as shown earlier in Fig. 5 (a). The thresholds of the considered algorithm are parametrized as follows: H = 200 - 230; S = 0 - 1; V = 0 - 1. If all three HSV-values lie outside of this intervals, the investigated pixel is classified as landable.

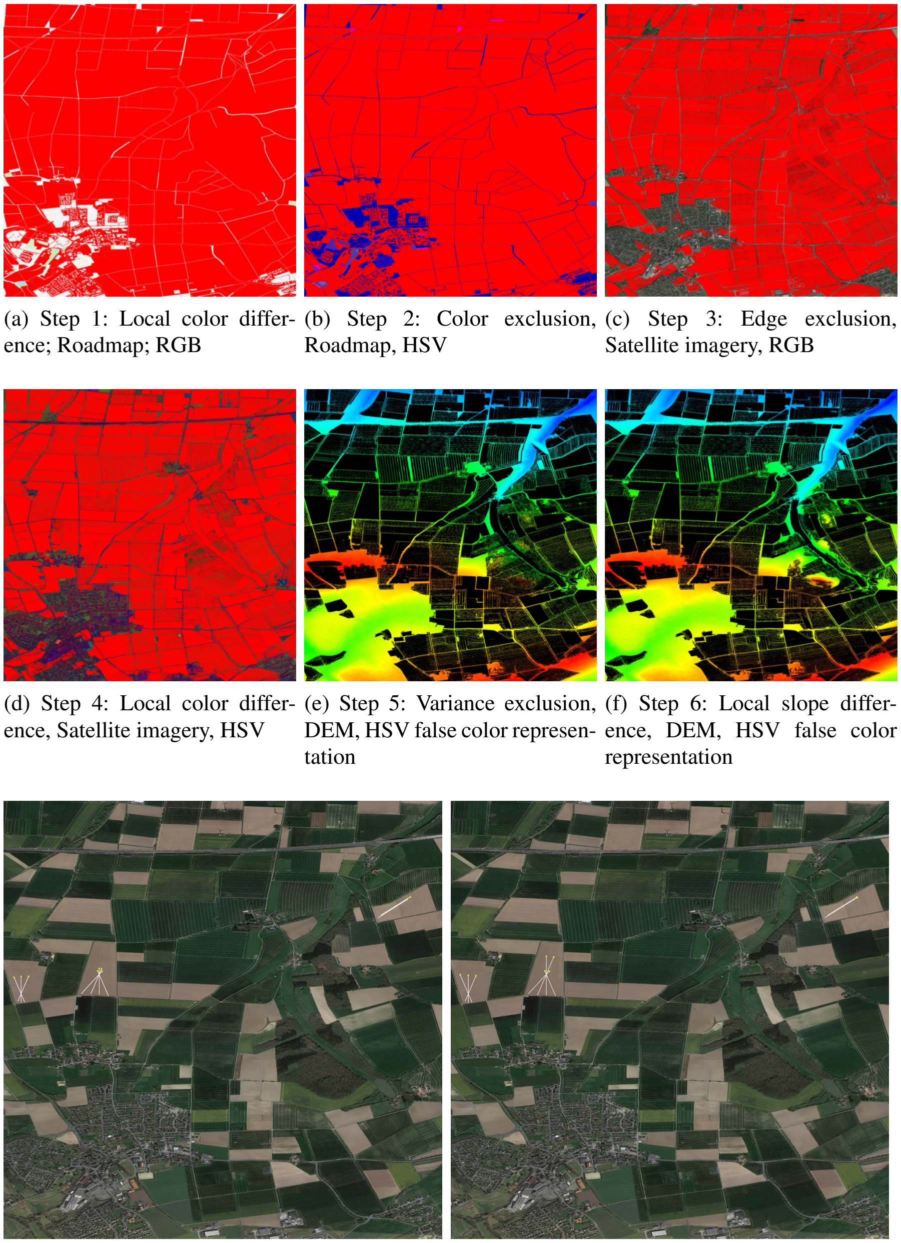
In Fig. 5 (c) the result of Canny edge detection is shown. We implemented a modified version to automatically determine the necessary thresholds. For the selection of the upper threshold, we applied Otsu's thresholding on the preliminary computed gradient map with the aim of the threshold selection which distinguish the two classes (edge or not) with the highest possible variance [20]. Afterwards, the selected upper threshold is divided by a factor of two, as recommended in [15], to set the lower threshold. It is obvious, that the number of homogeneous areas is reduced dramatically which will minimize the number of identified ELFs.

In Fig. 5 (d) the result of step four is presented. The three thresholds are defined as follows: H = 22; S = 1; V = 1. As a consequence of the parameterization only the hue value is considered during the execution of the local color difference method. If the hue of the neighboring pixel elements exceeds a delta of 22, the considered pixel is excluded and hence, marked as not landable.

In Fig 5 (e) the processing result of step 5 is shown. The black color denotes the segmentation result (suitable for an ELF). The other colors in the image indicate areas which are not landable. The maximum accepted variance is 0.001%. Otherwise, the considered area is assumed as too bumpy and tagged as unsuitable for an EL. It is obvious, that regions with fast changing elevation data are excluded and homogeneous areas are left for the further analysis.

The result of step six of the presented MMIPP is illustrated in Fig. 5 (f). The black color illustrates the segmentation result and denotes the landable areas. The slope difference was set to 10° so that only sharp elevation gradients are excluded from the remaining processing steps and classified as unsuitable for an ELF.

The search result of potential ELFs is shown in Fig. 5 (g). A total of nine potential ELFs were identified with an parameterization as follows: Minimum width = 32.67 m and length = 294.25 m; distance between two consecutive RPs = 294.25 m; sampling interval of ELF direction = 15°; maximum slope along sd = 1°, towards sd = 5° and orthogonal to sd = 3°.



(a) Step 1: Local color difference; Roadmap; RGB

(b) Step 2: Color exclusion, Roadmap, HSV

(c) Step 3: Edge exclusion, Satellite imagery, RGB

(d) Step 4: Local color difference, Satellite imagery, HSV

(e) Step 5: Variance exclusion, DEM, HSV false color representation

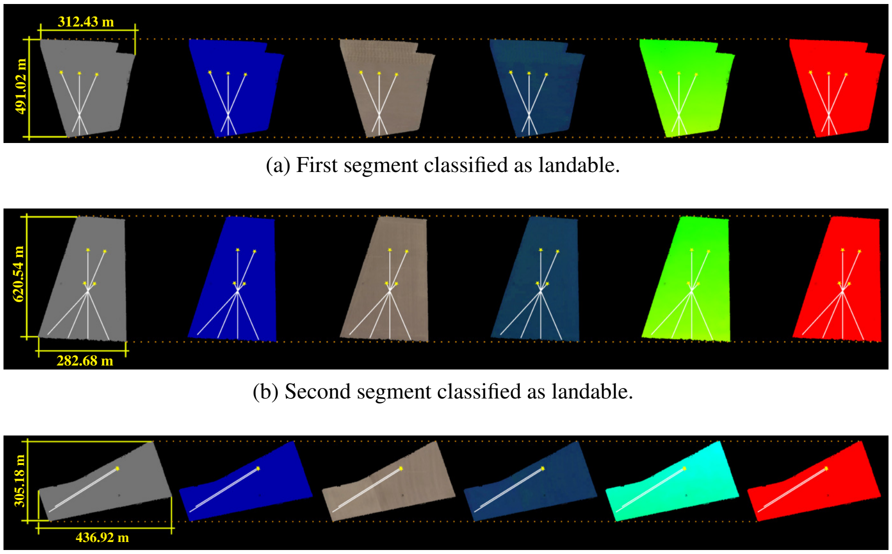
(f) Step 6: Local slope difference, DEM, HSV false color representation

(g) Step 7: Result of the potential ELF identification illustrated in RGB satellite imagery.

(h) Step 8: ELF evaluation and optimization result shown in RGB satellite imagery.

Fig. 5 Intermediate and final results of the MMIPP.

The recognized ELFs were further analyzed in the last step of the MMIPP to ensure the obstacle clearance of the final approach which yields to the result shown in Fig. 5 (h) The opening angle of the ELFs approach sector was 30°, the test distance was 2000 m and the glide angle was 5°. Apparently the final approaches of the identified ELFs are free of obstacles. The processing result is also shown in Fig. 6 where the mapping of the three input data layers, their transition in the HSV color space and the outcome of the developed MMIPP is illustrated. The ELFs are shown by white lines in each segment. The yellow stars denote the runway threshold. Furthermore, it can be seen that the first segment is 491.02 m long and 312.43 m width. The second segment extends to a length of 620.54 m and a width of 282.68 m. Lastly, the third segment is 305.18 m long and 436.92 m wide. Obviously, the dimensions of all considered segments are big enough for performing an EL with the investigated aircraft. The gray image segments in Fig. 6 (a) to (c) denote the roadmap data layer in RGB and the second segment in HSV color space. The third and fourth segment in Fig. 6 (a) - (c) show the corresponding satellite imagery in RGB and HSV color space. Furthermore, the fifth segment illustrates the false color representation of the digital elevation model after applying the bilinear interpolation and the last segment shows the final processing results of the considered areas. Fig. 6 depicts that the identified ELFs are placed in regions which are quite homogeneous in all analyzed



(a) First segment classified as landable.

(b) Second segment classified as landable.

(c) Third segment classified as landable.

Fig. 6 Mapping of all processed input layers, their corresponding transition into HSV color space and the final output represented by red color. First: RGB roadmap; Second: HSV roadmap, Third: RGB satellite imagery; Fourth: HSV satellite imagery; Fifth: HSV false color representation of DEM; Sixth: final segmentation result of the proposed MMIPP.

input data layers in both color spaces. Even the segments of the digital elevation model does only contain smooth changes regarding to slope.

Table 3 shows the position of each placed valid RP (RP lat/ RP lon). Thereby, values in rows colored with the same gray-level correspond to the same RP but have different EL directions. Furthermore, the optimized length of the ELF and increase in length with respect to the min. required dimension is represented (Δlength). By executing the proposed best permutation of the MMIPP a total of nine runways were identified. Moreover, four of the placed RPs were valid for identification of ELFs. Obliviously, the increase of the ELFs length – achieved by step 8 – ranges from 0.15% to 53.98% which depicts an extension from 0.44 m to 158.84 m. Unfortunately, the enlargement of the ELF in longitudinal direction is only evaluated regarding to slope and descent for the center line of the ELF. Therefore, the ELFs have to be further evaluated so that the criteria regarding bumpiness, slope and descent are still fulfilled after the enlargement of ELF length.

Table 3 ELF length and increase relating to the minimal required dimensions.

ELF idx	RP lat	RP lon	direction [°]	length [m]	Δ length [%]
1	51.53066	7.99056	210	295.72	0.5
2	51.52868	7.9875	210	326.2	10.9
3	51.52274	7.9587	60	294.69	0.15
4	51.52274	7.9587	90	446.85	51.86
5	51.52274	7.9587	245	453.09	53.98
6	51.52274	7.9587	225	296	0.6
7	51.52076	7.95069	60	325.66	10.68
8	51.52076	7.95069	90	311.6	5.89
9	51.52076	7.95069	45	301.79	2.56

The elevation data of the identified ELFs were extracted from our DM. For this purpose, the known properties of each ELF were used to determine their corresponding bounding box by applying the Bresenham line drawing algorithm proposed in [21]. Initially, the Bresenham algorithm was also applied to receive all elevation data points describing the ELF. Hence, a perpendicular line was drawn from each point on the transversal of the ELF boundary to the corresponding parallel point on the other transversal ELF boundary. This proceeding is only capable to deliver all contained elevation data points covered by the ELF aligned to the transversal and longitudinal direction as shown in Fig. 7. Obviously, some elevation points will be missed and others are extracted twice. For that reason, first the min. sized rectangle aligned to the x-axis is determined so that

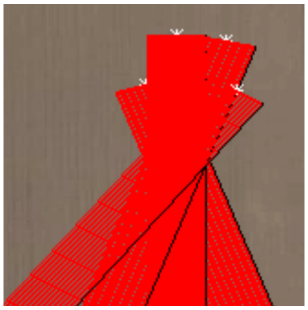
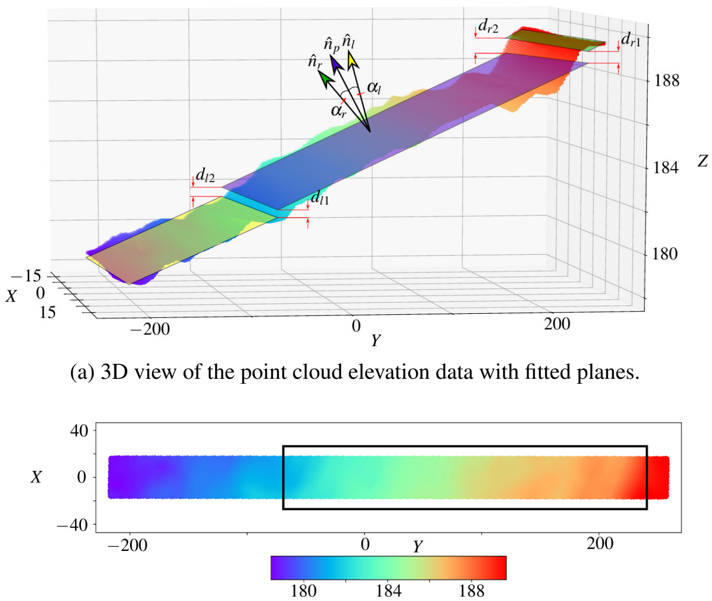


Fig. 7 Applied Bresenham line drawing algorithm for capturing the elevation data.

the whole dimensions of the considered ELF is contained. This enables the reduction of the computational demand regarding to the extraction of the elevation data of interest. For the decision which elevation data were covered by the preliminary determined bounding box of the ELF, the algorithm introduced in [22] is applied. Regrettably, this algorithm works non-reliable for the boundary points of the bounding box. Hence, the algorithm was adjusted with our prior acquired knowledge about the bounding box. To simplify the subsequent analysis of extracted elevation point cloud, each point is rotated by the direction of the ELF such that the whole point cloud is aligned to the *y*-axis.

For the evaluation of each length-optimized ELF, the data is partitioned into three parts – primal elevation data; The extension of the ELF towards and along the proposed approach direction – if the length extension in both directions has been executed and was preliminary considered as valid increase of length. Subsequently, the analysis of the generated point clouds is based on the fitting of a plane to each part of the ELF. Therefore, the least-squares minimization algorithm provided by the python library `SciPy` was used [23]. In Fig. 8 the evaluation result are illustrated. In Fig. 8a the calculated plane equations for the point clouds of an exemplary ELF



(a) 3D view of the point cloud elevation data with fitted planes.

(b) Two dimensional view of the elevation data.

Fig. 8 Evaluation procedure of the identified ELFs illustrated by an example

are illustrated. Each plane is depicted in a different color – violet: fitting to primal elevation data; yellow: fitting of elevation data enlargement to the left and green: fitting of elevation data enlargement to the right. Thereby, $\hat{n}_i$ with $i \in \{l, p, r\}$ denotes the normalized normal vectors of the three planes. The d_i with $i \in \{l1, l2, r1, r2\}$

is the relative distance between two consecutive planes with respect to the increase length of the ELF. The α_i with $i \in \{l, r\}$ depicts the angle between the normalized normal vectors, namely $\hat{n}_p$ and $\hat{n}_l$ as well as $\hat{n}_p$ and $\hat{n}_r$. The values of α_i and d_i are used to determine the suitability of the considered ELF for an EL. In Fig. 8b the 2D view of the elevation data of exemplary ELF is shown. The black rectangle highlights the dimension of the initial ELF. The color gradient is shown as smooth transition over the whole length of the ELF.

To facilitate the examination of the elevation data regarding to bumpiness, slope and descent proper thresholds for the maximum valid angle (α_i) and distance between two consecutive planes (d_i) has to be determined. For that reason, the dimensions of the considered aircraft are consulted. It has to be guaranteed that the aircraft avoids touching the ground except with the wheels. Therefore, the threshold for the maximum valid angle α_i between two consecutive planes has to be determined. Hence, the dimensions are obtained by the manual of the DA20-C1 where a sketch with a scale of about 1:100 is given as shown in Fig. 9 Mainly, it has to be assured, that the prop, tail and trunk are not touched by the ground during an EL. Because of the structure of the considered aircraft, it is assumed to be unlikely, that the trunk of this aircraft will touch the ground. Especially, the preliminary processing techniques had excluded such worse case scenarios. Therefore, the parameter α_3 in Fig. 9 is omitted during the following analysis. Furthermore, if the constraint of α_2 – shown in Fig. 9 – is violated, this may not be the case for α_1. Hence, α_2 is chosen as threshold value for the maximum valid angle between to consecutive fitted planes. For the Diamond DA20-C1 α_i is 11.89°. The maximum distance between two consecutive planes is determined by the size of the aircrafts nose wheel s_{nw}, the maximum valid angle α_i, the increase in length l_i and the steps size s_s as shown in Eq. 11

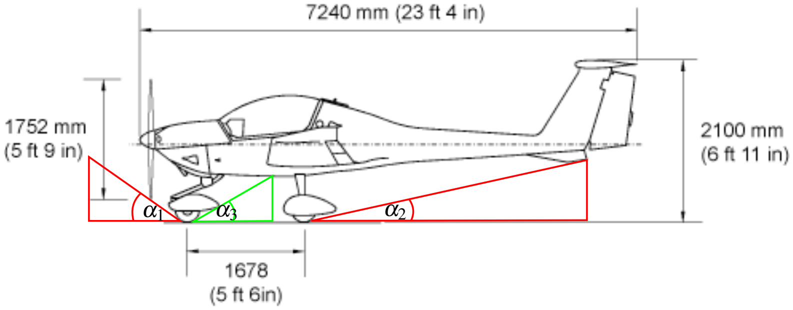


Fig. 9 Dimensions of the aircraft; critical angles denoted by red and green triangles [18, p. 1-5].

$$d_i = s_s \cdot \tan(\alpha_i) + \frac{s_{nw}}{s_{f1}} \cdot s_{f2} \cdot l_i \qquad (11)$$

where s_{f1} and s_{f2} are safety factors parametrized as follows: s_{f1} = 4 → we assume that the maximum angle conquerable by the nose wheel where it hits an obstacle is 45°; s_{f2} = 0.01 → per 1 m increase in the length of the ELF, the d_i value is raised by 1% of the quarter of the nose wheel size. Besides, s_s is assumed as 1 m and the s_{nw} is $5\frac{1}{4}$ in $\approx$ 13.335 cm as stated in [18, p. 1-6].

Table 4 shows the results of the application of Eq. 11 for d_r and d_l for ELFs which are increased in length greater than 1%. Thereby, the ELF indexes correspond to the one shown in Tab. 3 Additionally, the obtained actual value for the distance between two consecutive planes d_r^a and d_l^a is presented. The values of d_l^a for the ELFs with indexes four and seven exceed their corresponding thresholds d_l – highlighted in gray. For those ELFs the authors recommend to revert the length optimization along the left direction. Besides, it can be seen that the angle between two neighboring planes is below the preliminary mentioned threshold of α_i.

Table 4 Calculated thresholds for the distance between two consecutive planes and their actual values for ELFs with a increase in length greater than 1%.

ELF idx	Left/Middle			Right/Middle		
	d_r	d_r^a	α_r	d_l	d_l^a	α_l
2	1.59 m	1.59 m	2.20°	—	—	—
4	4.84 m	0.52 m	0.34°	0.51 m	1.14 m	0.23°
5	0.79 m	0.75 m	0.41°	5.14 m	0.46 m	0.19°
7	—	—	—	0.94 m	2.61 m	0.99°
8	0.71 m	0.29 m	0.76°	—	—	—
9	—	—	—	0.39 m	0.16 m	0.47°

In this section, the best permutation, parametrization and results of the introduced MMIPP were proposed. The selection of the MMIPP's configuration was mainly based on two criteria and the best ordering of the image processing techniques might be different from the chosen one depending on the considered area.

The results have shown that Canny edge might be better applied on a high resolution digital elevation model. This became obvious especially by processing fields in the satellite images which were covered by visible traces – caused by e. g. tractors. These traces were detected as edges also if the field might be suitable for an EL.

Due to the optimization of the ELFs length, further analysis was necessary to guarantee that the ELFs were still suitable for an EL. It was shown, that the optimization of the length should be reverted in two cases with respect to the selected thresholds and evaluation methods.

6 Conclusion and Future Works

In this paper a MMIPP was proposed to facilitate the identification of ELFs. Thereby, standard and self developed MV-techniques were introduced in Sec. 3 Those are subdivided into local and global exclusion methods. Afterwards, the DM with three data layers was introduced in Sec. 4. Besides, the bilinear interpolation was proposed which enabled the correct mapping of the elevation layer so that every pixel in the other two layers has a corresponding elevation value. In Sec. 5 the parametrization and results were proposed, further evaluated and discussed. First, the best permutation of the MMIPP was analyzed with respect to the execution time, the number of identified ELFs. Subsequently, the result of each image processing step as well as the final result of the proposed MMIPP for each analyzed data layer was presented. Each identified length optimized ELF was further examined and evaluated, if those are still suitable for an EL. Therefore, the thresholds for the chosen metrics were introduced and it was obvious that two optimizations of the ELFs length should be reverted.

In our future work we will investigate greater areas with a more precise digital elevation model provided for the region of North Rhine-Westphalia. Furthermore, the MMIPP will be enlarged with other machine vision and learning techniques. Beyond that, we will exchange our database by a `PostgreSQL` with its `PostGIS` database extender. It is worth to mention to facilitate the processing of bigdata by the usage of `Hadoop` or `Spark`. The proposed greedy search for placing RPs will be revised or completely exchanged by a other procedures. Moreover, the introduced MMIPP is incapable of real-time ELF selection and has to be done before the flight. To ensure the obstacle clearance of temporary objects (e. g. tractor, people etc.) of the ELFs real-time processing of on-board sensor data has to be performed.

Acknowledgment

The authors would like to thank Mr. Schneider for his contributions.

References

1. M. Klein, A. Klos, J. Lenhardt, and W. Schiffmann, “Moving target approach for wind-aware flight path generation,” *International Journal of Networking and Computing*, vol. 8, no. 2, pp. 351–366, 2018. [Online]. Available: http://www.ijnc.org/index.php/ijnc/article/view/189
2. L. Mejias, D. Fitzgerald, P. Eng, and X. Liu, “Forced landing technologies for unmanned aerial vehicles: Towards safer operations,” in *Aerial Vehicles*, T. M. Lam, Ed. Rijeka: IntechOpen, 2009, ch. 21. [Online]. Available: https://doi.org/10.5772/6481
3. T. Levora, B. Ondřej, and P. Pačes, “Emergency Landing Site Location using aerial Image Segmentation,” in *29th Congress of the International Council of the Aeronautical Sciences*, St. Petersburg, Russia, 2014, pp. 1–6. [Online]. Available: http://www.icas.org/ICAS_ARCHIVE/ICAS2014/data/papers/2014_0885_paper.pdf
4. L. Mejias and D. L. Fitzgerald, “A multi-layered approach for site detection in uas emergency landing scenarios using geometry-based image segmentation,” in *International Conference on Unmanned Aerial Systems*. Atlanta, Georgia: IEEE Control Society, 2013, pp. 366–372. [Online]. Available: https://eprints.qut.edu.au/60550/
5. M. Garg, A. Kumar, and P. B. Sujit, “Terrain-based landing site selection and path planning for fixed-wing uavs,” in *International Conference on Unmanned Aircraft Systems*, 2015, pp. 246–251.

6. F. Eckstein, B. Wittich, and W. Schiffmann, "Emergency landing field recognition based on elevation data using parallel processing." London: Digital Avionics Systems Conference, DASC, 2018.
7. L. Mejias, "Classifying natural aerial scenery for autonomous aircraft emergency landing," in *International Conference on Unmanned Aircraft Systems*, 2014, pp. 1236–1242.
8. S. T. Namin and L. Petersson, "Classification of materials in natural scenes using multi-spectral images," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1393–1398, 2012.
9. T. Y. Kim, G. Y. Sung, and J. Lyou, "Robust terrain classification by introducing environmental sensors," *8th IEEE International Workshop on Safety, Security, and Rescue Robotics*, 2010.
10. J. Chetan, M. Krishna, and C. V. Jawahar, "Fast and spatially-smooth terrain classification using monocular camera," *Proceedings - International Conference on Pattern Recognition*, pp. 4060–4063, 2010.
11. D. Fitzgerald, R. Walker, and D. Campbell, "A Vision Based Forced Landing Site Selection System for an Autonomous UAV," *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pp. 397–402, 2005. [Online]. Available: http://ieeexplore.ieee.org/document/1595612/
12. F. M. Rao, S. Aziz, A. Khalid, M. Bashir, and A. Yasin, "UAV Emergency Landing Site Selection System using Machine Vision," *Journal of Machine Intelligence*, vol. 1, no. 1, pp. 13–20, 2016.
13. Y. F. Shen, Z. U. Rahman, D. Krusienski, and J. Li, "A vision-based automatic safe landing-site detection system," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 294–311, 2013.
14. M. Warren, L. Mejias, X. Yang, B. Arain, F. Gonzalez, and B. Upcroft, "Enabling Aircraft Emergency Landings Using Active Visual Site Detection," in *Field and Service Robotics SE - 12*, ser. Springer Tracts in Advanced Robotics, L. Mejias, P. Corke, and J. Roberts, Eds. Springer International Publishing, 2015, vol. 105, pp. 167–181. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-07488-7_12
15. J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
16. Google Developers, "Map and tile coordinates," 2018. [Online]. Available: https://developers.google.com/maps/documentation/javascript/coordinates#tile-coordinates
17. J. C. Russ, *Image analysis and signal processing in electron microscopy. P. W. Hawkes, W. Owen Saxton, F. Peter Ottensmeyer and Azriel Rosenfeld (Editors). Published by Scanning Microscopy International, 1988, Chicago, IL, USA; ISSN 0892-953X*, vol. 18, no. 5. [Online]. Available: http://doi.wiley.com/10.1002/xrs.1300180513
18. *AIRPLANE FLIGHT MANUAL, DA20-C1*, DIAMOND AIRCRAFT INDUSTRIES INC., February 2013, rev. 27.
19. *AIRPLANE FLIGHT MANUAL, DA 40 D*, DIAMOND AIRCRAFT INDUSTRIES INC., December 2010, rev. 8.
20. N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979.
21. J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, Mar. 1965. [Online]. Available: http://dx.doi.org/10.1147/sj.41.0025
22. M. Shimrat, "Algorithm 112: Position of point relative to polygon," *Commun. ACM*, vol. 5, no. 8, pp. 434–, Aug. 1962. [Online]. Available: http://doi.acm.org/10.1145/368637.368653
23. "Optimization (scipy.optimize)," https://docs.scipy.org/doc/scipy/reference/tutorial/optimize.html, last accessed: 2018-10-20.