IC-EPSMSO

6th International Conference on Experiments/Process/System Modeling/Simulation/Optimization

Athens, Greece
8-11 July, 2015

Proceedings

Edited by

Prof. Dr. Demos T. Tsahalis, Director
Learning Foundation in Mechatronics (LFME)

VOLUME I
Pages 1-287

TABLE OF CONTENTS

LENGTH AND MANEUVER MINIMISATION OF UAS FLIGHT TRAJECTORY FOR PHOTOGRAMMETRIC MAPPING

Athanassios A. Stamos[1] and Dimitra I. Vassilaki[1]

[1]National Technical University of Athens (NTUA)
Iroon Polytechniou 9, GR-15780, Athens, Greece
e-mail: stamthan@central.ntua.gr, dimitra.vassilaki@gmail.com

Keywords: flight planning, optimisation, Unmanned Aerial Vehicle (UAV), Unmanned Aerial System (UAS), Remotely Piloted Aircraft Systems (RPAS)

Abstract. *Unmanned Aerial Vehicles (UAV), also known as Unmanned Aerial Systems (UAS) and Remotely Piloted Aircraft Systems (RPAS) are a modern tool in the toolbox of aerial photogrammetric mapping. One of the drawbacks of UAVs is that the area covered by an image is relatively small, and thus a large number of images are produced which lead to large photogrammetric processing cost. The limited capacity of the UAV battery is another drawback. This paper is about the optimisation of flights in order to have the minimum number of images to process and the minimum length of the actual flight path of the UAV. Large, with respect to the size of the image, areas with regular shape (not narrow, approximately convex) can be fitted with a grid of images. The optimal inclination angle of the grid which minimises the number of images is determined using a combination of the exhaustive search and divide-and-conquer algorithms. The extents of every row and column of the grid is dynamically computed for each angle. For irregular areas such as narrow zones centred by highways, the Simulated Annealing method (SA) is used to determine the position of every image. After the positions of the images are determined, the flight path which the UAV follows to collect every image is optimised using deterministic or stochastic algorithms. The method is tested with real world examples and gives encouraging results.*

1 INTRODUCTION

Photogrammetry is the science of making reliable and accurate measurements on photographs and images in order to create mapping products such as maps and 3D models of the surface of the Earth and other planets, and of the surface of salient objects of various sizes. Photogrammetry is generally classified into three broad categories: a) Close-range photogrammetry, b) aerial photogrammetry and c) satellite photogrammetry, depending on the distance of the sensor from the surface to be measured. In close-range photogrammetry the distance is less than 300 m, aerial photogrammetry covers the range of distances from an aircraft to the surface, and satellite photogrammetry covers the distance from orbit. Unmanned Aerial Vehicles (UAV), also known as Unmanned Aerial Systems (UAS) and Remotely Piloted Aircraft Systems (RPAS) are a modern tool in the toolbox of aerial photogrammetric mapping. They are classified as a method between terrestrial and aerial methods due to their limited height flight, and they exhibit the advantages and the disadvantages of terrestrial and aerial photogrammetry at the same time[1,2,3,4]. UAS have been used for many applications such as remote sensing and mapping[5], to capture micro-topography of Antarctic moss beds[6], tree stem mapping[7], and landslide monitoring[8], just to name a few.

One of the drawbacks of UAVs is that the area covered by an image is relatively small, and thus a large number of images are produced which lead to large photogrammetric processing cost. The limited capacity of the UAV battery is another drawback. This paper is about the optimisation of flights in order to have the minimum number of images to process and the minimum length of the flight path of the UAV. Very few related research papers are found in the literature. Qu[9] performed flight path optimisation using Voronoi diagrams for fast partial optimisation and genetic algorithms with Delaunay triangulation for slow full optimisation. In contrast this paper optimises both the division of the area to images and the flight path length.

The photogrammetric mapping of a non-trivial Area of Interest (AOI) needs many models which must span (cover) the whole area. A photogrammetric model is the common area of two consecutive overlapping images. The models themselves also overlap in order to accommodate aerotriangulation and compensate for possible error of the UAS navigation system. In effect, the Non-overlapping Area of the Model (NAM), which has rectangular shape, must span the AOI. Thus minimisation of the number of the images is in effect the

minimisation of the number of NAMs.

In practice there are two cases for the optimisation of the number of NAMs. In the first case the AOI has a regular shape (not very narrow, not very non-convex) and is much larger than the size of the image and the NAM. It follows that the length of the perimeter of AOI is small with respect to its area (Fig. 4). Naturally the NAMs must span the entire area and the design of their placement of NAMs is straightforward: each NAM should be placed next to each other without any overlapping in both X and Y dimensions, forming a grid (Fig. 4a). The whole area of all the NAMs is 100% utilised, except the NAMs placed at the AOI perimeter, where some fraction of the NAMs' area may be outside the AOI. Since the length of the perimeter of the regular AOI is small with respect to the area of the AOI, the number of NAMs on the perimeter in small and thus any optimisation of the placement of these NAMs (so that their area is more fully utilised) would have little effect on the overall number of NAMs and images. However the angle of the NAM grid with respect to the X-axis of the object coordinate system, does have an effect on the number of NAMs and an optimised value of the angle can be computed (Fig 4b).

In the second case the AOI has an irregular non-convex shape with large perimeter length with respect to its area such as narrow zones centred by highways (Fig. 6). In this case a significant number of images and NAMs are on the perimeter of the AOI and their areas are not fully utilised: on the average about half of the NAM's area is outside the AOI. Here the placement of the NAMs next to each other does not usually lead to the least number of images, and in fact the least number of images may be found by allowing the NAMs to overlap. This is not as strange as it may seem, since some area of the NAMs is already wasted (outside the AOI). This area can instead be wasted in the overlapping area. There is no systematic way to design the placement of the NAMs and in fact the number of possible configurations is astronomical, even if the placement of the NAMs is constrained to the vertices of a grid of a certain (small) step. For this reason the stochastic Simulated Annealing method[10] (SA) is used to compute the optimum placement of the NAMs. SA is a stochastic numerical optimisation method suitable for the optimisation of large scale problems and is able to find the global minimum among many local, poorer, minima. SA generates potential random changes to an initial set of image positions, which may be accepted or not. The fundamental characteristic of the method is that there is a finite probability to accept a change which increases the number of images instead of decreasing it. This probability diminishes over time, until no change is accepted and the optimum positions of the images are determined.

Once the number and the location of the NAMs are established, the UAS must visit the centre of each correspondent image in order to acquire it. The battery power of the UAS is limited and thus the length of the UAS flight path length must be the minimum possible in order to acquire as many images as possible. In the first case, where the NAMs (and the images) form a more or less regular grid, the optimum path is straightforward and is formed either visiting the grid by rows or by columns. In the second case the optimum flight path is unknown and the number of possible combinations of visiting the centre of N images is N!. Thus SA is employed to compute the optimum path.

The following part of the paper is organised as follows. Sections 2 and 3 present the placement of images and the flight path in regular areas. Sections 4 and 5 present the placement of images and the flight path on irregular areas. In section 6 examples for the two cases are presented and in section 7 conclusions and suggestions for further research are made.

2 PLACEMENTS OF IMAGES IN REGULAR AREAS

A regular AOI approximates a square, i.e. it is approximately convex and not very narrow, and it is much larger than the size of the image and the NAM. More formally, an AOI is regular when its perimeter is small with respect to its area and thus the number of NAMs on the perimeter is lower than the number of all NAMs which span the AOI:

$$N_{PER} \approx \frac{L_{PER}}{b_m} \;<<\; \frac{A_{PER}}{b_X \cdot b_Y} \approx N_{ALL} \tag{1}$$

where L_{PER} and A_{PER} are the length and the area of AOI, b_X and b_Y are the dimensions of the rectangular NAM, and b_m is the mean of b_X and b_Y or the mean size of the NAM. The NAMs which are inside the AOI are fully utilised, which means that their area is 100% inside the AOI and thus they can not be further optimised. On the other hand the NAMs on the perimeter are not fully utilised, as some of their area may be outside the AOI. Thus only the NAMs on the perimeter may be optimised, increasing the ratio of their area which is inside the AOI, and in effect reducing their number N_{PER}. However since N_{PER} is much lower than N_{ALL} the reduction of N_{PER} does not lead to dramatic reduction of N_{ALL} and thus it does not make much sense to employ a computationally expensive method like SA. A less expensive (and less efficient) method to divide the AOI into a

set of rectangular NAMs is straightforward. The AOI is divided into rows of height b_Y beginning at the lowest Y coordinate and ending at the highest Y coordinate of the AOI (which is assumed to be a possibly non-convex polygon or a collection of such polygons). Each row is divided into rectangles (NAMs) of width b_X (and height b_Y) beginning at the lowest X coordinate and ending at the highest X coordinate of the AOI within the current row. It is possible that some of these NAMs are completely outside the AOI. In order to find which NAMs to add to the set, the AOI is filled with a grid of points with step $0.1b_X \times 0.1b_Y$. If a NAM has at least 1 point in it, it is included in the set of NAMs of the AOI. The algorithm is shown in Fig. 1.

The algorithm divides an AOI to a set of NAMs parallel to the axes of the object coordinate system, which means that the angle φ between the width of the NAMs and the X-axis is zero. This is not obligatory at all and in fact a small and computationally inexpensive optimisation is to find which angle φ results to minimum number of NAMs. The deterministic minimisation method of golden section is employed for the optimisation[11]. The method needs a function which receives an angle between the angles φ_1 and φ_2 which bracket the optimum angle and returns the number of NAMs for this angle. However the algorithm shown in Fig. 1 assumes that the NAMs are parallel to the axes. Thus, instead of rotating the NAMs by angle φ, the coordinates of the AOI are rotated by $-\varphi$. After the optimum value of φ is determined by the golden section method, the NAMs are rotated by $+\varphi$. The angles φ_1, φ_2 which bracket the optimum angle can be found by exhaustive search. The algorithm is shown in Fig. 2.

Divide a polygonal AOI to a set of rectangular NAMs.
Create a grid S_P of points inside the AOI at every $dx=0.1b_X$ and every $dy=0.1b_Y$
Let $S_{NAM} = \{\}$ #The set of the NAMs inside the AOI
For each point P in S_P:
   Let kx = int(X_P / b_X)
   Let ky = int(Y_P / b_Y)
   Let X_{NAM} = kx · b_X
   Let Y_{NAM} = ky · b_Y
   If (X_{NAM} , Y_{NAM}) is not included in S_{NAM}:
      Add rectangle (X_{NAM}, Y_{NAM}, $X_{NAM}+b_X$, $Y_{NAM}+b_Y$) to S_{NAM}
   End if
End for
End

Figure 1. Algorithm to divide the AOI to rectangular NAMs

Divide a polygonal AOI to a set of rectangular NAMs using the optimum angle φ.
Let $S_\varphi = \{\varphi=0$ το π with step $\pi/60\}$
Let φ_{min} be the φ from S_φ which yields the least $f(\varphi)$
The optimum angle φ is bracketed between $\varphi_1=\varphi_{min}-\pi/60$ and $\varphi_2=\varphi_{min}+\pi/60$
Use golden section method with $f(\varphi)$, φ_1, φ_2 to compute angle φ_{OPT} which minimises number of NAMs
Rotate the coordinates of the polygon of the AOI by $-\varphi_{OPT}$
Compute the set of the NAMs S_{NAM} inside the AOI using the algorithm of Fig. 1
Rotate all NAMs of S_{NAM} by $+\varphi_{OPT}$
End

Function $f(\varphi)$ (returns the number of NAMs of a division rotated by φ).
Rotate the coordinates of the polygon of the AOI by $-\varphi$
Compute the set of the NAMs S_{NAM} inside the AOI using the algorithm of Fig. 1
Return the size of S_{NAM}
End

Figure 2. Algorithm to divide the AOI to rectangular NAMs inclined by optimum angle φ

---

3 MINIMUM FLIGHT PATH DISTANCE IN REGULAR AREAS

When the AOI is regular the division of the AOI into rectangular NAMs resembles a regular grid (Fig. 4). If the grid is perfectly regular (exactly a square or rectangle with no discontinuities), then the flight path which yields the minimum flight length is to visit each row or column one by one in order. If the grid is not perfectly

rectangular there is room for optimisation. However in the case of regular AOI, the grid is almost regular and any optimisation will not be dramatic and thus it does not make much sense to employ a computationally expensive method like SA. Instead, an inexpensive and deterministic optimisation method is to try the following flight paths and keep the one which yields the least flight length:

a. Visit the first row left to right, the second right to left, the third left to right and so on until the last row.

b. Visit the first row right to left, the second left to right, the third right to left and so on until the last row.

c. Visit the first column bottom to top, the second top to bottom, the third bottom to top and so on until the last column.

d. Visit the first column top to bottom, the second bottom to top, the third top to bottom and so on until the last column.

4 PLACEMENT OF IMAGES IN IRREGULAR AREAS

When the AOI is irregular a significant number of the images (and NAMs) are on the perimeter of the AOI and their areas are not fully utilised (Fig. 6). On the average about half of the NAM's area is outside the AOI, and thus in theory the number of NAMs could be halved, but there is no straightforward way to place the NAMs and accomplish this. Specifically the placement of the NAMs next to each other does not usually lead to the least number of images, and in fact the least number of images may be found by allowing the NAMs to overlap. This is not as strange as it may seem, since some area of the NAMs is already wasted (outside the AOI). This area can instead be wasted in the overlapping area. There is no systematic way to place the NAMs and in fact the number of possible configurations is astronomical, even if the placement of the NAMs is constrained to the vertices of a grid of a certain (small) step. For this reason SA is used to compute the optimum placement of the NAMs, which is a general method of optimising large scale problems under arbitrary constraints where the optimum value is hidden among many local poorer minima.

4.1 Simulated annealing

In order to optimise (minimise) the value a desired property of a problem, SA starts with an initial configuration of the problem (for example an initial placement of the NAMs). To this initial design the method makes random trial modifications. A modification is always accepted if it decreases the property value. However, if a modification increases the value (negative change), there is a real probability that it will also be accepted, according to the Metropolis formula[12]:

$$P(\Delta t, T) = e^{-\Delta \tau / T} \quad , \quad \Delta t = t_2 - t_1 > 0 \tag{2}$$

where t_1 and t_2 are the property values before and after the modification. The property value t is the analogue of energy in real metallurgy annealing, and T is the analogue of temperature which expresses the agility of the modifications. The value of the initial temperature T is chosen so that (2) gives a good chance for negative changes to be accepted. The number of trial random modifications should be large, and it usually taken as the dimensionality of the problem multiplied by a large constant. The fact that negative changes are accepted may seem counter-intuitive, but it is precisely this which enables the method to avoid being locked to local minima[11]. The temperature is gradually decreased in a series of steps and the process is repeated for each temperature step. Decreased temperature in (2) leads to the acceptance of fewer negative modifications. This gives a natural criterion to terminate the method. When no modifications, either positive or negative, are accepted in a given temperature, the optimum configuration has been determined. It must be noted, that this is not the absolute minimum in the mathematical sense, since the method is stochastic. However, it is very close to it. Any constraints of the problem may be enforced implicitly by adding a penalty to the property value if the constraint is not fulfilled or it is partially fulfilled. As the method minimises the property values, the violations of constraints are eliminated.

In order to apply the simulated annealing method many aspects and parameters of the method must be adapted to the problem being solved. The aspects and parameters described below were determined through experimentation.

4.2 Initial configuration

The initial placement of the NAMs is trivial. The method begins with no NAMs at all, and relies on the random modifications to add NAMs into appropriate places.

4.3 Trial random modifications

The type of random modifications is crucial to the method as they should be able to lead to the optimum solutions. In this paper the following types of modifications were used:

a.   Addition of an NAM. A NAM is added to the set of NAMs. The coordinates of the lower left corner of

NAM are randomly chosen so that they are within the enclosing rectangle of the AOI, extended by the size of one NAM in all directions. In order to reduce the number of possible positions and thus speed up the method, the candidate positions of the NAMs are limited to integer multiples of $b_X/4$, $b_Y/4$ (Fig. 6b). For the same reason, the maximum number of NAMs is also limited to 1.1N, where N is the number of NAMs determined by the method of Section 2, and the coefficient 1.1 gives SA some room to try negative modifications.

b.   Deletions. A random NAM of the set of NAMs is deleted.

c.   Relocation. A random NAM of the set of NAMs is relocated to a new random position. The new position is determined as described in additions.

When the method creates a random modification, the type of the modification (a), (b) or (c) is randomly chosen, taking into account that for (b) and (c) the set of NAMs must not be empty and for (a) the number of NAMs must be less than 1.1N.

4.4   Penalty

The method minimises the number of NAMs on the condition that the NAMs span the AOI. This can not be ensured by the random modifications described earlier, and thus a penalty is added to the number of NAMs which is proportional to the area of the AOI that is not covered by the NAMs. The area not covered by NAMs is hard and time consuming to compute in the case of non-convex AOI. Thus the AOI is filled with a set of grid of points S_P with step $0.1b_X \times 0.1b_Y$. For every point of S_P that is not inside any NAM a penalty of 0.1N is added to the number of NAMs, where N is described in paragraph 4.3a. The minimisation of the number of NAMs plus the penalty ensures that the method will try to eliminate the penalty and thus ensure the NAMs span the AOI.

4.5   Temperature steps

The temperature starts with an initial value which is then gradually decreased in a series of steps. For each temperature step, $10N_C$ trial modifications are made, where N_C is the current number of NAMs in the set (the dimensionality of the problem). The temperature is decreased by a constant factor in each successive step: $T_{i+1} = 0.95\,T_i$

The initial temperature should allow many negative modifications (which increase the number of NAMs). This is translated as a big probability, for example P=0.95 (or 95%) in the Metropolis formula (2). If we knew Δt we could solve (2) for T. In order to estimate Δt, before the beginning of the method we make 100N trial modifications and take the average of all Δt which are produced. Then T is estimated by solving (2).

4.6   Algorithm

The algorithm implementing the simulated annealing method is shown on Fig. 3.

5   MINIMUM FLIGHT PATH DISTANCE IN IRREGULAR AREAS

When the AOI is irregular a significant number of the images (and NAMs) are on the (irregular) perimeter and there is no obvious way to visit each one so that the total flight length is minimum. In fact the problem is identical to the famous "travelling salesman problem[11]", where one should try all the possible combinations and select the combination which yields the minimum flight length (exhaustive search). Since the possible combinations are $N_{NAM}!$ (factorial of N_{NAM}) this is impractical for $N_{NAM}>14$ or so. Thus the stochastic simulated annealing method is also applied here as described in Numerical Recipes[11].

6   APPLICATION

The methods described in this paper were implemented in Python[13] and were integrated into ThanCad[14], an open-source CAD, for user friendliness and ease of use. The methods were applied to a regular and an irregular area.

6.1   Regular area

The area of interest shown in Fig. 4 is a subset of the mapping which was performed for a segment of the Egnatia highway in northern Greece. It has roughly rectangular shape of dimensions 2000×650 m and it is partly non-convex. The AOI will be surveyed by a modern UAS capable of taking images of size 817×613 m.

```
Simulated annealing optimising the number of NAMs.
Compute the set of the NAMs S_NAM inside the AOI using the algorithm of Fig. 1
Let N be the size of S_NAM
Create a grid S_P of points inside the AOI at every dx=0.1b_X and every dy=0.1b_Y
Let S_NAM = {}                                    #Set the initial configuration empty
Let N_P,OUT be the number of points of S_P not in any rectangle of S_NAM
Let t_1 = size(S_NAM) + N_P,OUT * 0.1 * N         #Energy=number of NAMs plus penalty
For i=1 to 100N:
        Make random modification to S_NAM (add, delete or relocate a NAM)
        Let N_P,OUT be the number of points of S_P not in any rectangle of S_NAM
        Let t_2 = size(S_NAM) + N_P,OUT * 0.1 * N    #Energy=number of NAMs plus penalty
        Let Δt = t_2-t_1
        Let t_1=t_2
End For.
Compute average Δt
Compute initial T
Let S_NAM = {}                                    #Set the initial configuration empty
While True:
        Let m=0
        For i=1 to 10*size(S_NAM):
                Make random modification to S_NAM (add, delete or relocate NAM)
                Let N_P,OUT be the number of points of S_P not in any rectangle of S_NAM
                Let t_2 = size(S_NAM) + N_P,OUT * 0.1 * N    #Energy=number of NAMs plus penalty
                Let Δt = t_2-t_1
                If Δt < 0:
                        Save (positive) modification to S_NAM
                        Let t_1=t_2
                        Let m = m + 1
                Else:
                        Compute probability P=exp(-Δt/T)
                        If P < probability of negative changes so far:
                                Reject modification        #Do not save modification to S_NAM
                        Else:
                                Save (negative) modification to S_NAM
                                Let t_1=t_2
                                Let m = m + 1
                        End If
                End If
        End For
        If  m = 0: Exit While
        Let T = 0.95 T
End While
S_NAM has the optimum NAMs
End Simulated Annealing.
```

The overlap of the images is 80% in both directions and thus the NAM is a rectangle of size 164×123 m. Fig. 4a shows the division of the AOI to horizontal NAMs using the deterministic algorithm summarised in Fig. 1. The method produced 96 NAMs, 38 of which are on the perimeter and have 50% utilisation on the average. In Fig. 4b the same AOI was divided into inclined NAMs of the same size. The optimum inclination angle (14.1 deg) was computed with the deterministic method of golden section summarised in Fig. 2. The method produced 89 NAMs, 36 of which are on the perimeter. The utilisation of the NAMs on the perimeter is clearly better than before, resulting to about 7% reduction of the number of NAMs, which is modest but at practically no cost as the computation took seconds in an interpreted computer language[15] and 9 years old hardware.
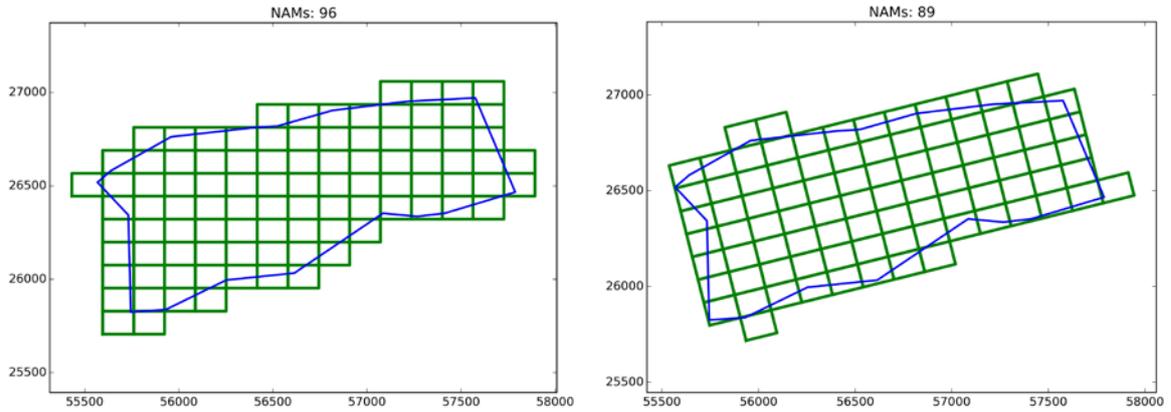
Figure 4. Regular area (blue) divided into rectangles (green) using deterministic algorithms: a. horizontal rectangles (left), b. inclined rectangles (right)

Once the configuration of the NAMs was determined the flight path which minimises the flight length was computed using the deterministic method described in Section 3. All the 4 possible flight paths were tried: visiting the centres of the NAMs by rows left to right or right to left, or by columns bottom to top or top to bottom. The results are shown in Fig. 5. The flight path with the minimum flight length is when the UAS visits the NAMs by columns and begins from top to bottom. Fig. 5 shows that the other 3 flight paths contain at least one leg which is considerably larger that the other legs. It is also evident that the rectangular shape of NAMs affects the flight length, i.e. it is probably better to fly parallel to the small side of the NAMs.
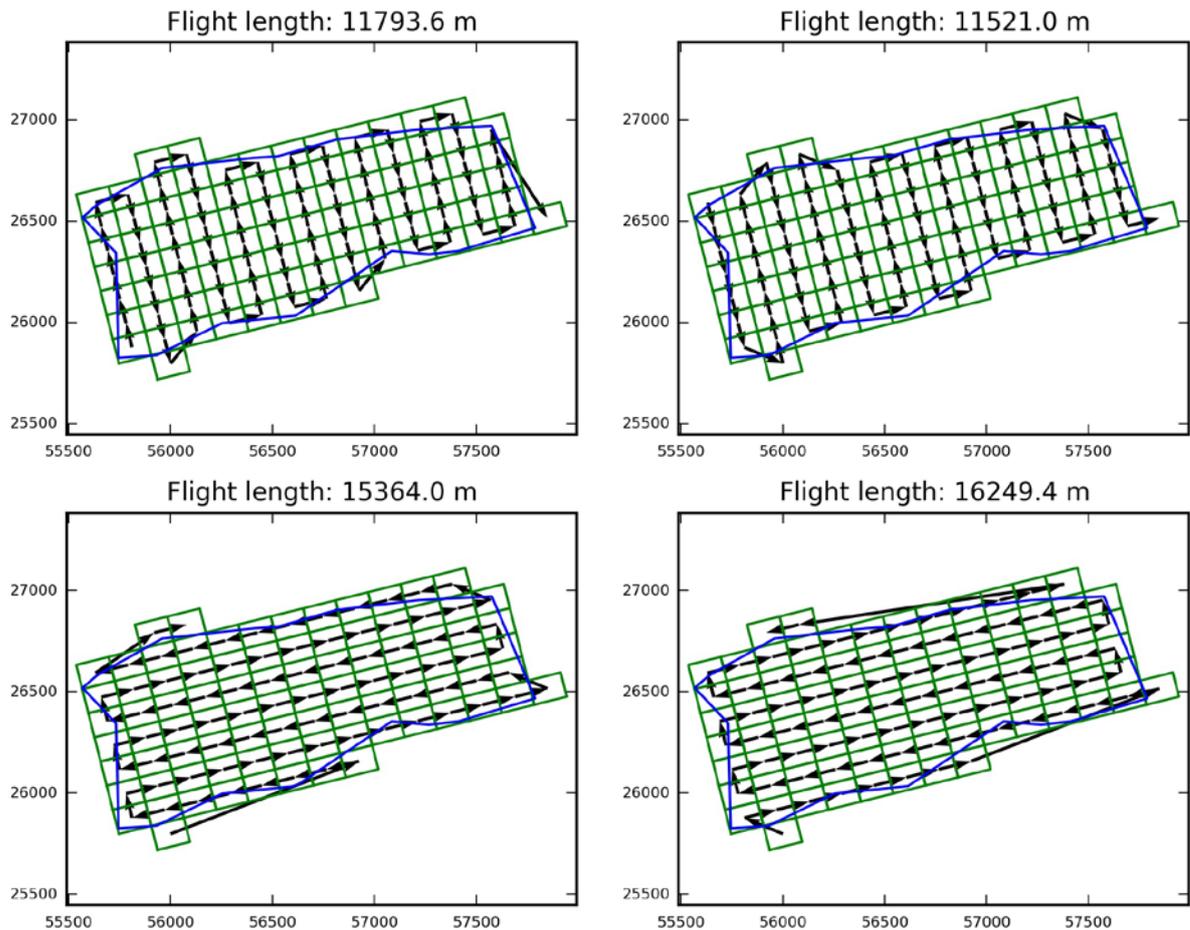
Figure 5. Regular area (blue) divided into rectangles (green): deterministic computation of minimum flight length (black); four flight paths are tested

6.2 Irregular area

The area of interest shown in Fig. 6 is for a small rural road and is about 2800 m long and 150 m wide. The NAM is again a rectangle of size 164×123 m. Fig. 6a shows the division of the AOI to horizontal NAMs using the deterministic algorithm for regular areas summarised in Fig. 1. The method produced 52 NAMs, all on the perimeter, which have less than 50% utilisation on the average. In Fig. 6b the same AOI was divided into NAMs using SA as summarised in Fig. 3. As the method is stochastic, each time it is applied to a problem, it produces slightly different results and a common practice is to repeat the computation 3-5 times and take the best result. The method produced 44 NAMs, all on the perimeter. The utilisation of the NAMs is clearly better than before, resulting to about 15% reduction of the number of NAMs. Fig. 6b shows that the reduction was possible because the NAMs were allowed to overlap. The computation took more than an hour in an interpreted computer language and 9 years old hardware.

Once the configuration of the NAMs was determined the flight path which minimises the flight length was computed using the stochastic simulated annealing method as described in Section 5. As the method is stochastic, each time it is applied to a problem, it produces slightly different results and a common practice is to repeat the computation 3-5 times and take the best result. The flight path which yields the minimum flight length is shown in Fig. 7a. The flight path shown in Fig. 7b contains a cyclic path of the perimeter and yields slightly larger flight length. Thus, at least in this irregular area, and perhaps contrary to common sense, visiting the perimeter (cyclically) does not lead to the minimum flight length.
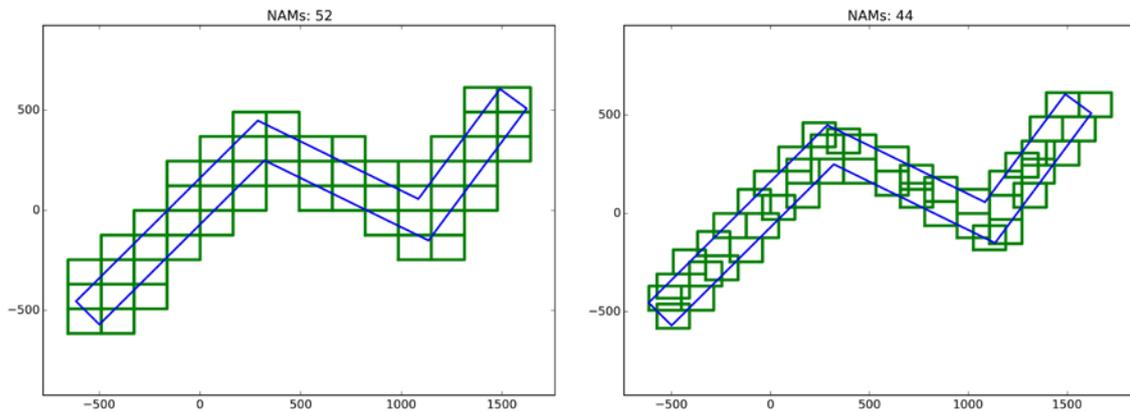
Figure 6. Irregular area (blue) divided into horizontal rectangles (green): a. deterministic algorithm (left), b. stochastic method
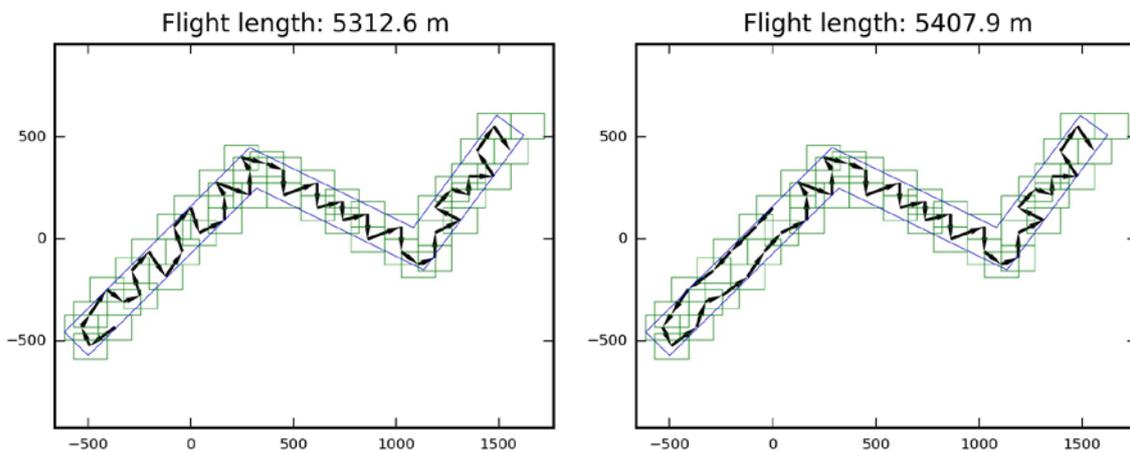
Figure 7. Irregular area (blue) divided into rectangles (green). Stochastic computation of minimum flight length (black): a. Minimum flight length, b. Slightly larger flight path

7 CONCLUSIONS

Two methods for the division of the area of interest to rectangular images were presented, one for regular and

one for irregular AOIs. Additionally, two methods for the computation of the optimum flight path for regular and irregular AOIs were presented. The methods were successfully applied to real world cases of mapping and led to fewer images and lesser flight length than straightforward methods. Suggestions for future research are the combination of the methods, the application of stochastic methods to large areas using supercomputers, and taking into account the recharge of the batteries of the UAS.

ACKNOWLEDGMENTS

REFERENCES

[1] Eisenbeiß, H. (2009). *UAV photogrammetry*. Zurich, Switzerland: ETH.

[2] Colomina, I., & Molina, P. (2014). "Unmanned aerial systems for photogrammetry and remote sensing: A review", ISPRS Journal of Photogrammetry and Remote Sensing, 92, 79-97.

[3] Remondino, F., Barazzetti, L., Nex, F., Scaioni, M., & Sarazzi, D. (2011), "UAV photogrammetry for mapping and 3D modeling – Current status and future perspectives", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 38(1), C22*.

[4] Haala, N., Cramer, M., Weimer, F., & Trittler, M. (2011), "Performance test on UAV-based photogrammetric data collection", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1/C22), 7-12.

[5] Everaerts, J. (2008), "The use of unmanned aerial vehicles (UAVs) for remote sensing and mapping", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37, 1187-1192.

[6] Lucieer, A., Turner, D., King, D. H., & Robinson, S. A. (2014), "Using an Unmanned Aerial Vehicle (UAV) to capture micro-topography of Antarctic moss beds", *International Journal of Applied Earth Observation and Geoinformation*, 27, 53-62.

[7] Fritz, A., Kattenborn, T., & Koch, B. (2013), "UAV-based photogrammetric point clouds—Tree stem mapping in open stands in comparison to terrestrial laser scanner point clouds", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 40, 141-146.

[8] Rau, J. Y., Jhan, J. P., Lo, C. F., & Lin, Y. S. (2011), "Landslide mapping using imagery acquired by a fixed-wing UAV", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38, 1-C22.

[9] Qu, Y. H., Pan, Q., & Yan, J. G. (2005), "Flight path planning of UAV based on heuristically search and genetic algorithms". *Industrial Electronics Society, 2005. IECON 2005. 31st Annual Conference of IEEE*.

[10] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983), "Optimization by Simulated Annealing", Science, Vol. 220, pp. 671–680.

[11] Press, W.H., Flannery, B.P., Teukolsky, S.A. and Vetterling, W.T. (1992), *Numerical Recipes in FORTRAN: The Art of Scientific Computing*, 2nd ed., Cambridge University Press, Cambridge.

[12] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953), "Equations of State Calculations by Fast Computing Machines", Journal of Chemical Physics Vol. 21, pp. 1087–1092.

[13] Python Software Foundation (2015), *Python Language Reference*, version 3.4. Available at http://www.python.org, Last updated Jun 2, 2015.

[14] Stamos, A.A. (2007), "ThanCad: a 2dimensional CAD for engineers", *Proceedings of the Europython2007 Conference*, *Vilnius, Lithuania, July 9-11*.

[15] G. van Rossum, *Python tutorial*, Technical Report CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.