SIMULATING UNMANNED-AERIAL-VEHICLE BASED LASER SCANNING DATA FOR EFFICIENT MISSION PLANNING IN COMPLEX TERRAIN

M. Bremer [1,2,] *, V. Wichmann [3], M. Rutzinger [2], T. Zieher[2], J. Pfeiffer[2]

[1] Institute of Geography, University of Innsbruck, Innsbruck, Austria – magnus.bremer@uibk.ac.at
[2] Institute for Interdisciplinary Mountain Research, Austrian Acadamy of Sciences, Innsbruck, Austria – magnus.bremer@oeaw.ac.at, martin.rutzinger@oeaw.ac.at, thomas.zieher@oeaw.ac.at, jan.pfeiffer@oeaw.ac.at
[3] Laserdata GmbH, Innsbruck, Austria – wichmann@laserdata.at

Commission II, WG II/10

KEY WORDS: Unmanned Aerial Vehicle based Laser Scanning, ULS, LiDAR, Simulation, Mission Planning

ABSTRACT:

In complex mountainous terrain the mapping efficiency is a crucial factor. Unmanned aerial vehicle (UAV) based laser scanning (ULS) has the capability for efficient mapping, as it allows realizing higher flight velocities, higher flying altitude above ground level (AGL) and larger distances between neighbouring flight strips, compared to image based techniques. However, fully utilising the efficiency of the system in mission planning (especially for complex terrain projects, where occlusions and differently inclined surfaces are present) is prone to miss the project requirements in terms of point density and strip overlap. Therefore, the numerical simulation of point densities is a helpful tool for realizing a reliable planning of scan coverage. We implemented a ray-tracing-based ULS-simulator, specifically designed for emulating the mechanism of a Riegl VUX-1LR laser scanner carried by a Riegl RiCOPTER. The simulator can consider copter and scanner motion, which makes it possible to generate synthetic scan data excluding or including the aircraft movement due to aerodynamics by using either planned trajectories from a flight planning software or recorded and post-processed trajectories from an inertial measurement unit (IMU). Laser shots are simulated by intersecting rays from the virtual scanner with a mesh-based digital surface model (DSM). The results show that the tool generates plausible synthetic laser point distributions. However, this is only the case, when aircraft aerodynamics are considered, as the effect of striping due to flight control corrections during the flight is very prominent. It can be shown that applying the presented tool for mission planning (without knowing the actual flight movements) has to consider an error margin of $\pm 50 pts/m^2$ in order to guarantee a compliance with the planned project requirements. Nevertheless, the consideration of terrain by a high resolution DSM, especially in complex terrain, improves the correlation between simulated and real point densities significantly.

1. INTRODUCTION

Unmanned aerial vehicle (UAV) based laser scanning (ULS) is a powerful technique in order to efficiently map project areas of up to a few km^2. Compared to UAV based photogrammetry (i.e. structure-from-motion and dense matching approaches), a point measurement does not require at least two observations, which allows a higher flexibility in planning of overlaps. It allows realizing higher flight velocities, higher flying altitude above ground level (AGL) and larger distances between neighbouring flight strips. ULS has the advantage of airborne laser scanning (ALS) in terms of scanning perspective, the penetrability of vegetation and its independence from direct project area accessibility. Finally, the wide field of view (FOV) of ULS systems additionally contributes to their mapping efficiency.

In complex mountainous terrain the mapping efficiency is a crucial factor and has to be optimized in order to guarantee reasonable field logistics. This causes a high risk of erroneous mission planning, leading to incomplete coverage, unsatisfying point densities and strip overlaps. To overcome this, an adequate planning of strip configurations, flying heights and scanning parameters, including pulse repetition rate (PRR) and angular scan resolution, are required.

For simple scan scenes with flat terrain and planar surfaces, this can be done analytically by using the formulas given by e.g. Baltsavias (1999). Thus, point densities and predicted footprint diameters can be computed as a function of the planned scanning configuration. For this planning task specialized software applications such as the RiParameter Tool (Riegl, 2019) are available.

However, in complex mountainous terrain the acquisition geometry is not constant during the flight. The structure of the terrain leads to strong variations in flying altitude AGL. Additionally, prominent terrain features can occlude lower regions behind. Thus the analytical prediction of point densities in such project areas can only be seen as a rough estimate. Hence, the point coverage rather needs to be predicted with a numerical model, simulating scanner and platform movements and radiative transfer by ray tracing.

A variety of tools for the numerical simulation of laser scanning data exists (e.g. Lovell et.al. 2005, Lohani and Mishra 2007, Kim et al. 2009, Kukko and Hyyppä 2009, Hodge 2010, Bechtold and Höfle 2010, Gschwandtner, et al 2011, Bremer et al. 2017, Bremer et al. 2018). While the simulation principles are similar, some tools are designed for specific platforms only (airborne, terrestrial), while others make use of more generalized parameterization in order to handle different platform types (e.g. Bechtold and Höfle 2010). The latest advances can be seen in the simulation of beam divergence and

* Corresponding author

physically correct full-waveform signal recording (Kukko and Hyyppä 2009) and the implementation of full 3D simulation scenes (e.g. Kim et al. 2009).

Most of the simulation tools have been applied for simulating laser scanning data on procedurally modelled scenes or based on synthetic or planned flight trajectories. However, for efficient flight planning the simulation tools need to be able to handle the data of real scenes, which allows the comparison of the simulated scanning result with the real flight and scan data. Additionally, a pre-flight simulation (planned flight trajectories) and a post-flight simulation (flown trajectories) are required in order to decompose effects of aircraft movement (path corrections and variations in angles of attack due to wind effects), effects of the scan pattern and effects of local terrain features. In order to use flown trajectories for simulation, a tight integration of global navigation satellite system (GNSS) data and inertial measurement unit (IMU) data with the tool's file import interface is needed.

In order to optimize mission planning in mountainous regions, we implemented a custom-tailored simulation tool for ULS simulation and tested its capabilities for the prediction of point densities. As local point densities and strip overlaps are the most crucial parameters in planning, we used a simple ray simulation, not considering multi echoes and full-wave-form recording.

2. METHODS

2.1 Template Device and Programming Environment

We implemented a simulation tool, able to predict point densities and coverage in complex terrain for UAV mission planning (if an a-priori digital surface model (DSM) is available). As a template device we used our Riegl VUX-1LR system (Riegl, 2019) with an Applanix AP-20 inertial measurement unit (IMU) (Applanix, 2019) and the Riegl RiCOPTER as carrier platform. The template device uses a rotating mirror, deflecting the laser pulses orthogonally to the longitudinal axis of the instrument. This causes a spiral scan pattern around the longitudinal axis and allows a field of view (FOV) of 336 degree (Fig.1a). Pulse repetition rates (PRR) of 50 – 820kHz can be realized.

The simulation tool is implemented in C++ as a plugin of the software SAGA-GIS (Conrad et al. 2015). SAGA allows the handling of traditional GIS 2.5D raster and 3D vector formats but also includes a native point cloud format. Thus it can handle a whole 3D simulation scene in a georeferenced way. This makes it possible to directly compare real geo data such as point clouds or raster maps with simulated datasets.

2.2 Input Layers

The tool requires two basic input layers: 1) a (polygonal) DSM given as a mesh in 3D shape file format and 2) a planned or flown trajectory given as a 3D line shape file or as SAGA point cloud. The *planned* trajectories are given as a set of straight lines (coming from the flight planning software, UgCS 2019), where only the start and end point (X,Y,Z) are defined. These are stored as 3D line shape files. Due to flight conditions the *flown* input trajectories show undulations affecting both position and orientation of the device during the realization of the planned straight flight line. The *flown* trajectories are an ordered set of positions (X,Y,Z) with associated orientations (φ, θ, ψ) and are stored in point cloud format, showing a temporal resolution of 0.005s. The simulation tool handles both trajectory datasets as flight paths, where a pair of two consecutive vertices defines a flight path segment (Fig.1a,b).

2.3 Input Parameters

The input parameters include the planned configuration of the scanning device and the flight parameters:

PRR = Pulse repetition rate [kHz]
φ_{scan} = Total scan angle, FOV [336°]
δ_φ_{scan} = Angular resolution of laser scanner [°]
v = Flight speed [m/s]
δ_X_{lever} = Lever arm x [m]
δ_Y_{lever} = Lever arm y [m]
δ_Z_{lever} = Lever arm z [m]
(the lever arm describes the distances between IMU coordinate system (IMUCS) and scanner-own coordinate system (SOCS))

2.4 Simulating Movements of Platform and Device

In order to iteratively simulate the rotation of the scanner mirror and the movement of the UAV platform along the given flight trajectory, the step length (*sl*) along the path, between two consecutive laser shots, has to be determined (Eq. 1).

$$sl = v/PRR \tag{1}$$

For each shot the instantaneous mirror rotation ($inst\_\varphi_{scan}$) and UAV position ($\overrightarrow{inst\_pos}$) have to be defined (algorithm 1). This is done by continuously increasing the path distance on the trajectory (*path_on_segment*) by *sl* and increasing the angle $inst\_\varphi_{scan}$ by δ_φ_{scan}. This is done for n steps (Fig. 1). By stepping along the given trajectory segment using the *path_on_segment* variable, the ($\overrightarrow{inst\_pos}$) is computed.

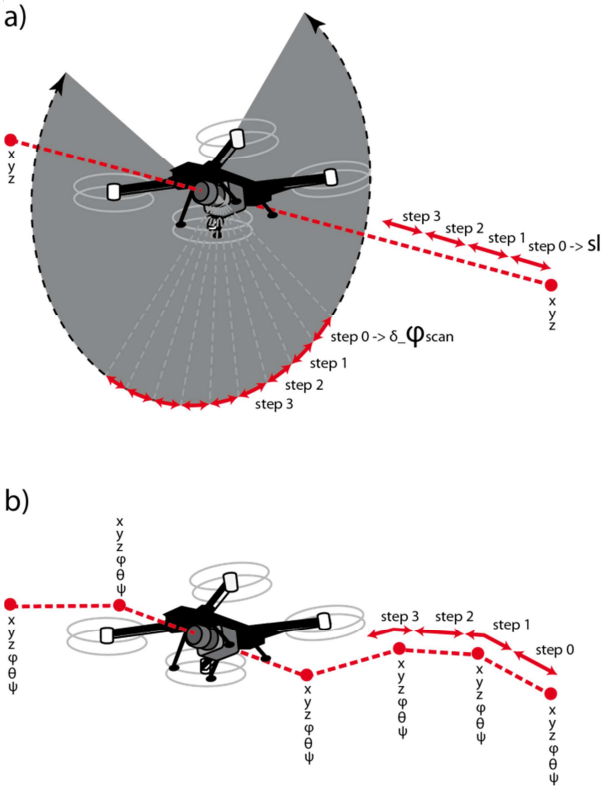

Figure 1. Principle of motion simulation for a) *planned* trajectory without orientation information per vertex and for b) *flown* trajectory with orientation information per vertex

Algorithm 1. Pseudo code of motion simulation

```
path_on_segment = 0
inst_φscan = 0
for segment in trajectory do
        v0 = get_first_vertex( )
        v1 = get_second_vertex( )

        # get unit vector for segment
        dir = (v1 − v0)/|v1 − v0|

        while path_on_segment ≤ |v1 − v0| do
           inst_pos = v0 + dir * path_on_segment

           φ, θ, ψ  =compute_platform_orientation( )

           shot, shot_o = get_shot(inst_φscan,φ,θ,ψ,inst_pos)

           Raytracing (shot, inst_pos)

           path_on_segment += sl # step along
           inst_φscan += δ_φscan      # angle step
        end while

        # compute remaining path distance to continue
        # on next segment
        path_on_segment += sl -|v1 − v0|
end for
```

Besides the instantaneous position, the instantaneous orientation of the UAV has to be computed (Algorithm 1: compute_ platform_orientation()). Therefore the values for roll, pitch and yaw have to be derived (φ, θ, ψ). For planned trajectories, complex flight movement can't be considered (Fig. 1a). No side wind is assumed leading to a roll of zero and a yaw aligned to the flight direction. As a multicopter, dependent on the flight speed, the Riegl RiCOPTER applies an angle of attack using the pitch control in order to generate a forward movement. This is affecting the scanning direction (backward looking) and can be defined by the user. Thus the parameters are:

Φ = roll[°]= 0°]
θ = pitch[°] (defined by user)
ψ = yaw[°] (defined by 2D orientation of trajectory segment)

For flown trajectories the orientations for each segment vertex are known (v0 = Φ0, θ0, ψ0; v1 = Φ1, θ1, ψ1) (Fig. 1b). These real orientations can show differences to the planned ones as the copter's flight controller compensates drift forces due to side winds and turbulence by applying angles of attack on the roll and pitch controls. While a compensation using the roll control (rotation around the longitudinal axis of the aircraft) leads to no significant change in the scan pattern, a correction on the pitch control (rotation around the transversal axis) leads to rhythmic increases and decreases in point density along the flight path.
The orientation at the instantaneous position is defined as follows.

$$\Phi = \left(\frac{\Phi 1-\Phi 0}{|\overrightarrow{v1-v0}|*path\_on\_segment}\right) + \Phi 0 \qquad (2)$$

$$\theta = \left(\frac{\theta 1-\theta 0}{|\overrightarrow{v1-v0}|*path\_on\_segment}\right) + \theta 0 \qquad (3)$$

$$\psi = \left(\frac{\psi 1-\psi 0}{|\overrightarrow{v1-v0}|*path\_on\_segment}\right) + \psi 0 \qquad (4)$$

If ψ0 is < 5° degree and ψ1 > 355° (or vice versa), a special case is defined and the latter value is substracted by 360°.

For the computation of the instantaneous shot direction (algorithm 1: get_shot()), all parameters have to be combined for each iteration of the simulation. The initial shot direction and origin ($\overrightarrow{shot}$, $\overrightarrow{shot\_o}$) given in scanner own coordinates (Fig. 2a), has to be transformed by the given inst_φ_{scan}, lever arm (x,y,z), φ, θ, ψ and inst_pos (x,y,z) into a global coordinate system(GLCS) (Fig. 2a-c).

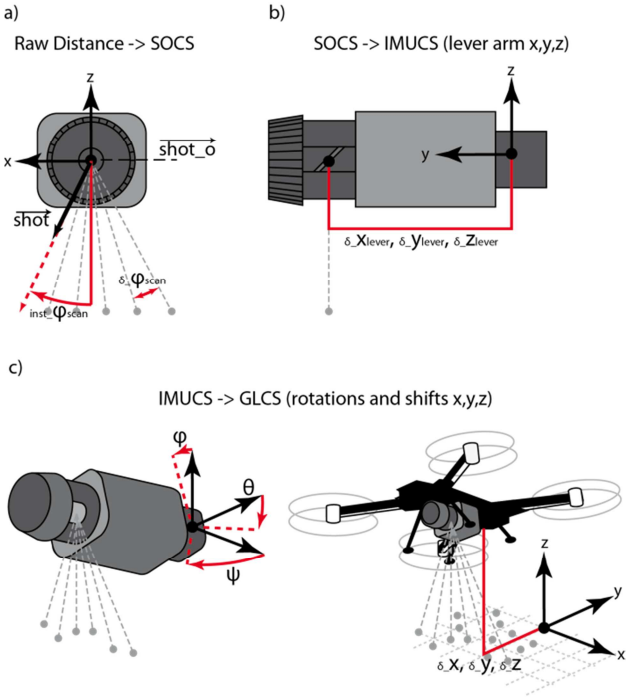

Figure 2. Different transformation components for the transformation of $\overrightarrow{shot}$, $\overrightarrow{shot\_o}$ from SOCS to GLCS. a) scan angle with respect to SOCS, b) lever arm from scanner centre to IMUCS, c) rotations and shifts between IMUCS and GLCS

The given transformation components are defined as follows:

$$\overrightarrow{shot\_o} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix} \quad (5) \qquad \overrightarrow{shot} = \begin{bmatrix}0\\0\\-1\\1\end{bmatrix} \quad (6)$$

$$m\varphi_{scan} = \begin{bmatrix}\cos(\text{inst\_}\varphi_{scan}) & 0 & sin(\text{inst\_}\varphi_{scan}) & 0\\ 0 & 1 & 0 & 0\\ -sin(\text{inst\_}\varphi_{scan}) & 0 & \cos(\text{inst\_}\varphi_{scan}) & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad (7)$$

$$m\varphi = \begin{bmatrix}\cos(\varphi) & 0 & sin(\varphi) & 0\\ 0 & 1 & 0 & 0\\ -sin(\varphi) & 0 & \cos(\varphi) & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad (8)$$

$$m\theta = \begin{bmatrix}1 & 0 & 0 & 0\\ \cos(\theta) & -sin(\theta) & 0 & 0\\ sin(\theta) & \cos(\theta) & 0 & 0\\ 0 & 0 & 0 & 1\end{bmatrix} \quad (9)$$

$$m\psi = \begin{bmatrix} \cos(\psi) & -sin(\psi) & 0 & 0 \\ sin(\psi) & \cos(\psi) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

$$m_{lever} = \begin{bmatrix} 1 & 0 & 0 & \delta\_X_{lever} \\ 0 & 1 & 0 & \delta\_Y_{lever} \\ 0 & 0 & 1 & \delta\_Z_{lever} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

$$m_{shift} = \begin{bmatrix} 1 & 0 & 0 & \text{inst\_pos\_X} \\ 0 & 1 & 0 & \text{inst\_pos\_Y} \\ 0 & 0 & 1 & \text{inst\_pos\_Z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

The basic transformation in order to transform the instantaneous shot origin and direction from SOCS to GLCS is given in Eq. 13 and 14, considering Eq. (5-12). Thereby, the shot vector becomes updated. This transformation has to be performed for each simulated shot, while the parameters are iteratively modified through the aircraft and scanner movement.

$$\overrightarrow{shot\_o} = m_{shift} * m\psi * m\theta * m\varphi * m_{lever} * \overrightarrow{shot\_o} \quad (13)$$

$$\overrightarrow{shot} = m\psi * m\theta * m\varphi * m\varphi_{scan} * \overrightarrow{shot} \quad (14)$$

2.5 Preprocessing of DSM mesh

Before the numerical simulation starts, an auxiliary voxel structure is built based on the DSM mesh (Fig. 3). The total size of the voxel grid is defined by the 3D extent of the mesh. The voxel size is set to 0.5x0.5x0.5 m by default. A 3D Bresenham algorithm (Bresenham 1965) is used in order to find all voxel cells intersecting with a given mesh triangle. After this, the unique identifier of the associated triangle is mapped to the respective voxel cell. This is done for all mesh triangles and leads to lists of triangle IDs mapped to each voxel cell. If a voxel cell shows no intersection with a mesh triangle, the cell entry keeps empty. This auxiliary structure is used for improved ray tracing. This structure is not as efficient as bounding volume hierarchies (BVH) but allows a combination with other voxel based data representations such as turbid media for the description of porous objects such as vegetation.

2.6 Ray Tracing

For each iteration of the basic algorithm, a single shot is simulated. This is done by computing the intersections of the ray, given by the shot direction and origin, and the 3D bounding box of the voxel structure. Using the line segment, which is defined by the given intersection points, the algorithm traces along the line segment through the voxel structure (grey cells, Fig. 3). If a visited voxel cell is not empty, all triangles mapped to this voxel cell are tested for an intersection. If the line intersects with a triangle, the exact intersection point is stored as a simulated laser point.

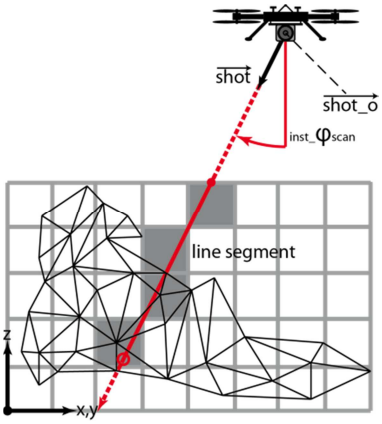

Figure 3. Principle of ray tracing, using an (instantaneous) input shot vector ($\overrightarrow{shot}, \overrightarrow{shot\_o}$) and an input mesh. Gray rectangles show the principle of the auxiliary voxel structure used for ray tracing

3. EXPERIMENTS

3.1 Simple Terrain - Model Airfield

In order to have a controlled test environment, we tested the simulations and comparisons of the results in a flat model airfield scene. A single flight strip was acquired with a flight speed of 8 m/s, a PRR of 820 kHz and an angular resolution of 0.0285°. The flying altitude AGL was 80 m. The lever arm was set to x=0.019, y=0.1776, z=0.0004. The θ angle for the planned trajectory was set to -10°. The given trajectory is shown in Fig. 4. For the conducted flights we simulated scan data with the same scanner configuration for both the planned (straight line) and the flown trajectories (including undulations due to aerodynamics). For the simulation a DSM with 10 cm resolution was derived from a regional ALS data set.

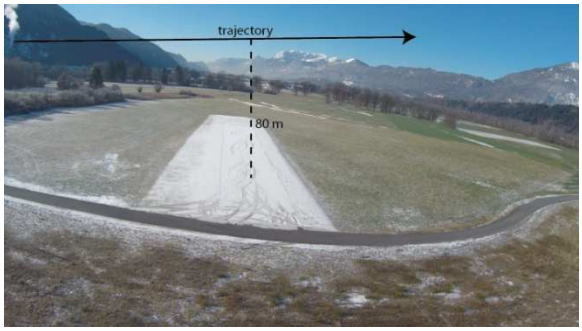

Figure 4. Flat terrain test case at model airfield

3.2 Complex Terrain – Deep-seated Landslide Slope

For complex terrain, we used a single scan strip from a larger landslide monitoring project (Fig. 5 and 6). The average flying height AGL for this test case was 70m with a minimum of 50 and a maximum of approximately 100m. The flight speed was 8 m/s, the PRR was 820 kHz and the angular resolution was 0.0476°.The simulations were conducted in the same way as for the simple test case. The 10 cm DSM was derived from a terrestrial laser scanning (TLS) campaign conducted one week before the ULS campaign.

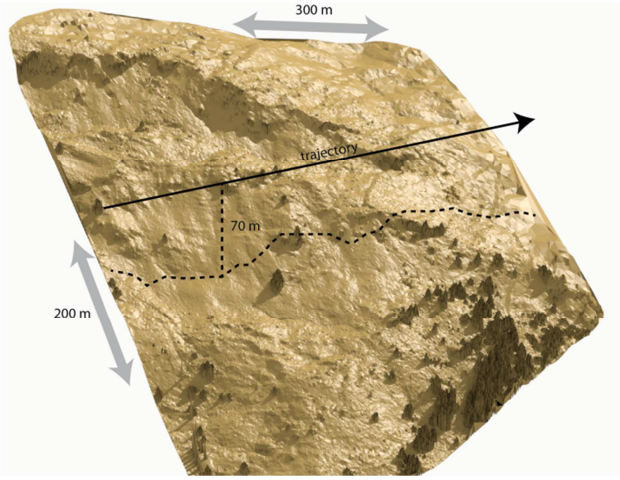

Figure 5. Polygonal DSM of the complex test area, used for simulation

Figure 6. Setting up the system in complex terrain

4. RESULTS

For the simulated and real point clouds we computed 1 m resolution raster maps and counted the laser points per cell (Fig. 7 and 9). Additionally, we generated scatterplots describing the relationship between real point densities and simulated point densities (Fig. 8 and 10). The first visual impression of the real point density patterns shows a striping effect orthogonal to the flight trajectory. When the flown trajectory with known path undulations was used for simulation, this striping effect could be recreated synthetically for both test cases (Fig 7b and Fig. 9b). As a result the correlation between simulated and real point densities shows high coefficents of determination (R^2) (Fig 8a and 10a). For the simple test case, R^2 (0.98) is higher than for the complex terrain test case (0.93).

As the planned trajectories assume a constant roll, pitch and yaw angle and constant flight speed for the given flight path, no striping is visible in the simulated data. Such striping effects caused by aerodynamics are not predictable. The simulations of the planned trajectories show ideal point density patterns with high point densities in the close nadir areas and lower point densities in the areas further away. The relationship between scan range and point density shown in Fig. 8c (blue dots) makes this ideal distribution apparent. The striping effect of the real data causes a stronger scatter of the point densities distribution. However, the point densities follow the same trend (Fig. 8c, red dots). The scatterplot also shows that for planned trajectories an error margin of approx. 50 pts/m^2 has to be considered. Thus, in the worst case, the real point densities can be 50 pts/m^2 lower than planned.

In the complex terrain test case the planned trajectory simulation allows a description of the point density patterns related to terrain features. This includes i) the overall trend in point density distribution, ii) high point density spots and iii) occlusion patterns, which are all simulated correctly. However, the striping effects and the terrain based patterns interfere with each other, which leads to a weak R^2 for the simulated point densities from planned trajectories (0.27 for the model airfield; 0.43 for complex terrain). For the simple test case, this effect is even stronger.

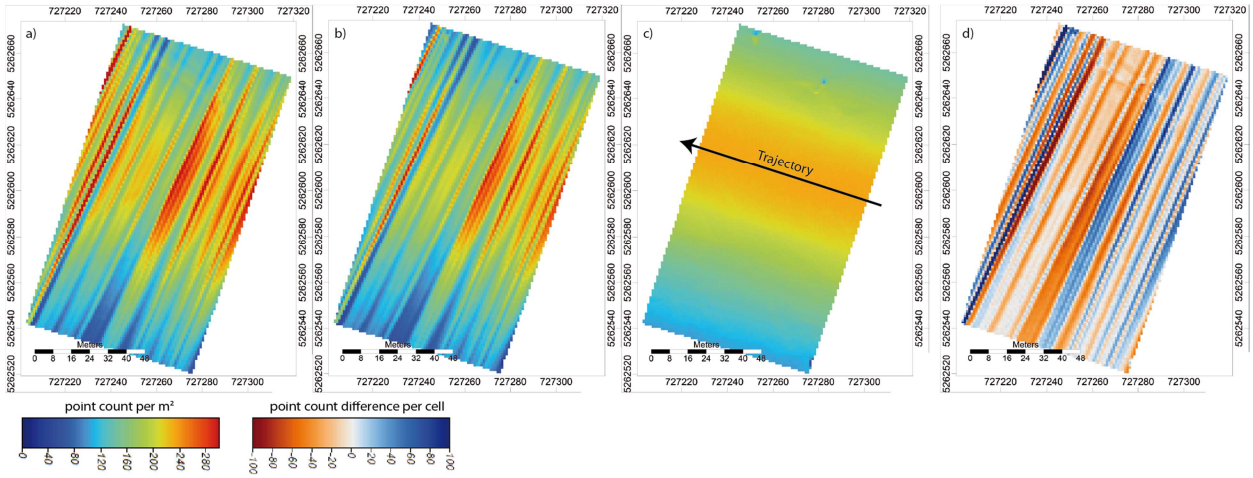

Figure 7. Laser point density and density difference maps of simple test case (model airfield): a) real point densities of the acquired point cloud; b) simulated point densities based on flown trajectory with known copter behaviour; c) simulated point densities based on planned trajectory; d) difference between real and simulated (planned) point densities

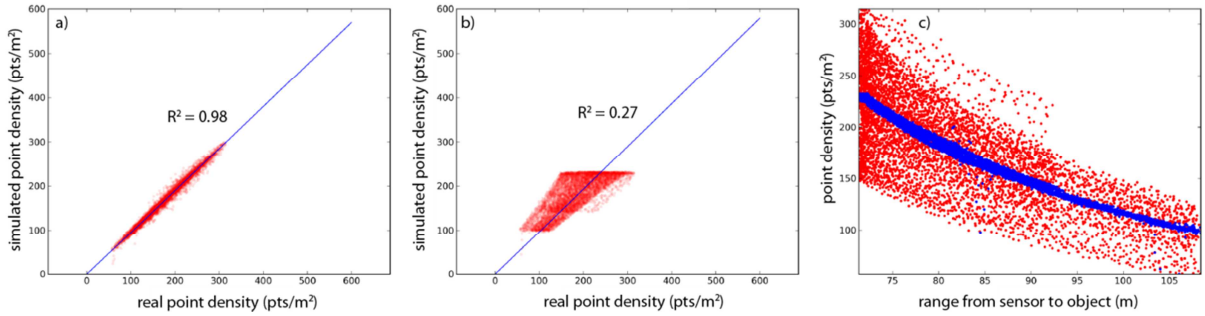

Figure 8. Comparison of real and simulated point densities for model airfield: a) scatterplot of the relationship between real point densities and simulated point densities (flown trajectories); b) scatterplot of the relationship between real point densities and simulated point densities (planned trajectories); c) scatterplot of the relationship between scan range and point density for simulated data (planned trajectory = blue and real data = red)

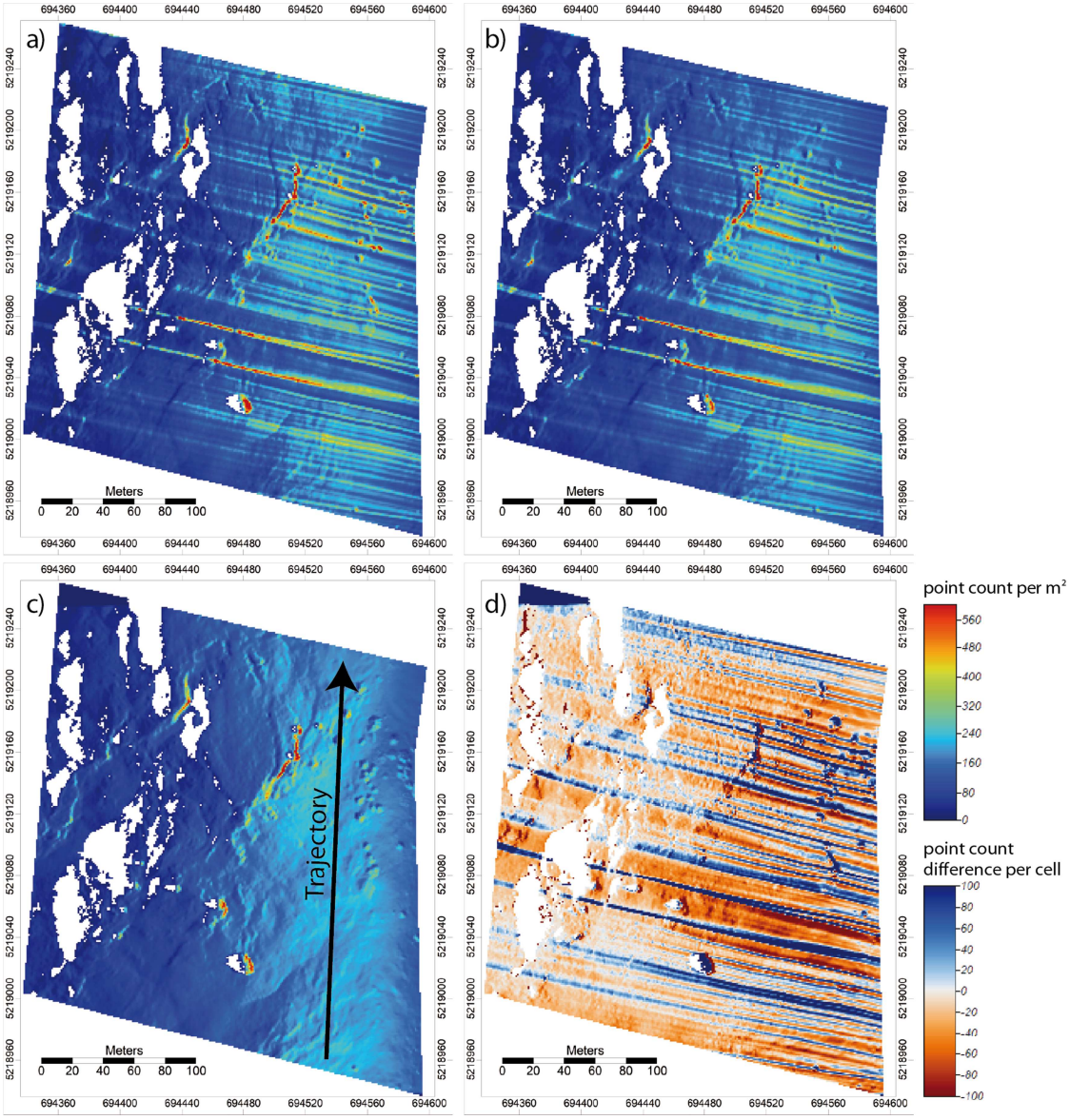

Figure 9. Laser point density and density difference maps for the complex terrain test case: a) real point densities of the acquired point cloud; b) simulated point densities based on flown trajectory with known orientations and positions; c) simulated point densities based on planned trajectory; d) difference between real and simulated (planned) point densities

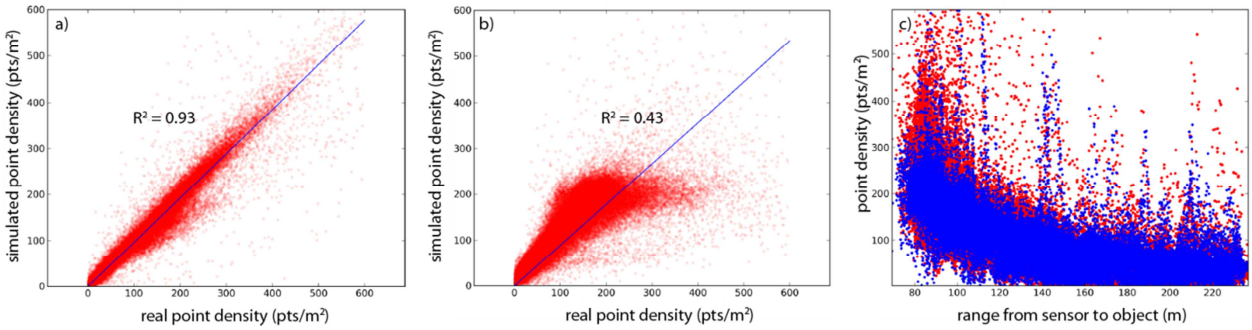

Figure 10. Comparison of real and simulated point densities: a) scatterplot of the relationship between real point densities and simulated point densities (flown trajectories); b) scatterplot of the relationship between real point densities and simulated point densities (planned trajectories); c) scatterplot of the relationship between scan range and point density for simulated data (planned trajectory = blue and real data = red)

5. DISCUSSION

5.1 Performance of the Simulation Model

The simulation model shows good performance in recreating ULS point densities both in simple and complex terrain. This is especially true for the simulation with flown trajectories as this significantly reduces the number of unknown factors influencing the scanning mechanism. This supports the correctness of the implemented scanning principle and its parametrization.

The simulated and real data sets show slight differences, which can be seen in the given R^2 values. This can be explained by three facts: i) the limited detail of the used 0.1 m DSMs, is not able to describe micro scale roughness, including shrubs and grasses growing on the ground surface. ii) the simulated laser points do, in no case, coincide with the real laser point locations. Due to the described mechanism of scanner motion simulation, the actual mirror position during runtime depends on the start position of the scanner-recording on the path and the step length. Thus the simulated point positions were simulated (almost) at random and are independent from the real point positions. iii) the method of point computation is very simplified in the used simulation mode. As described above, only simple triangle-ray-intersections are used for point positioning, ignoring the effects of beam-divergence, as realized by other authors.

However, even with the applied simple point simulation mode, it is possible to generate synthetic point distributions close to the real ones.

5.2 Usability of the Simulation Model for Mission Planning

The results show, that the application for mission planning, with planned straight trajectories, is limited. The prediction of point densities by using planned trajectories for simulation differs significantly from the real point densities. This is due to the prominent striping effect because of aircraft dynamics, which cannot be predicted with satisfying accuracy in a simulation model. In mission planning only straight flight paths are constructed, as the real conditions for the flight and the resulting undulations are not known in advance. Nevertheless, it is crucial for the operator of the mission to know that the mission requirements can be fulfilled irrespective of the actual conditions during flight.

Despite the lack of knowledge about platform behaviour, this can only be realized by considering a reasonable error margin for flight planning with straight trajectories. For the given examples, the unpredictable effect of aerodynamics shows an error margin of approximately ± 50 pts/m^2. For a conservative flight planning, the nominal value of point density could be increased by this value.

Except for the striping effects, the simulation tool performs well in predicting varying point densities on differently inclined terrain surfaces. All structures that are important for the prediction of strip overlaps and point distributions such as occluded areas and hot spots of point density in steep terrain can be predicted correctly. An interesting result is that in complex terrain the influence of striping is reduced in relation to the influence of the terrain. Thus, point density patterns related to terrain features with differently inclined and occluded surfaces become more important. In the point density maps of the complex terrain test case the terrain features are clearly visible for real data, flown trajectory simulation and planned trajectory simulation. For the flat terrain test case, the terrain effect is negligible and the striping effect becomes the outstanding factor for point density determination. This explains, why the R^2 of the relationship between real and planned trajectory simulation is lower for the flat terrain test case than for the complex terrain test case. This in turn pronounces the importance of complex terrain consideration for the simulation of point densities.

6. CONCLUSION

The simulation tool provides an added value for mission planning. Considering an error margin to compensate platform movement effects, an adequate estimation of point densities on differently inclined surfaces and with increasing scan distances is possible. The location of occluded regions in particular could be predicted with satisfying correctness.

Besides mission planning, the tool is also useful for post mission analysis, which is demonstrated by the correct simulation of aerodynamic effects when recorded flight trajectories of the real flights are used. By comparing pre- and post-flight simulation, single factors of influence can be decomposed for a detailed analysis of the flight.

In general it can be summarized that for ULS scan flights aircraft motion has a strong effect onto the resulting point density distributions, which has to be considered in detail. For flat terrain this effect exceeds the effects of terrain inclination and range but becomes attenuated in complex terrain where the influence of different terrain features becomes more pronounced. A suitable way of analysing this phenomenon is the use of a simulation tool as presented in this paper.

ACKNOWLEDGEMENTS

We thank the Austrian Bundesministerium für Bildung, Wissenschaft und Forschung for financing the ULS infrastructure project 4DLamb. The data acquisition was done within the Horizon 2020 OPERANDUM project (GA 776848) investigating the potential of nature-based solutions for mitigating hydro-meteorological risks.

REFERENCES

Applanix, 2019. www.applanix.com. Last accessed: 16 January 2019.

Baltsavias, E. P., 1999. Airborne laser scanning: basic relations and formulas. *ISPRS Journal of Photogrammetry and Remote Sensing* 54 (2-3), 199-214.

Bechtold, S., Höfle, B., 2016. Helios: A multi-purpose LiDAR simulation framework for research, planning and training of laser scanning operations with airborne, ground-based mobile and stationary platforms. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Volume III-3, XXIII ISPRS Congress, Prague, Czech Republic.

Bremer, M., Wichmann, V., Rutzinger, M., 2018. Multi-temporal fine-scale modelling of Larix decidua forest plots using terrestrial LiDAR and hemispherical photographs. *Remote Sensing of Environment* 206. 189-204.

Bremer, M., Wichmann, V., Rutzinger, M., 2017. Calibration and validation of a detailed architectural canopy model reconstruction for the simulation of synthetic hemispherical images and airborne LiDAR data. *Remote Sensing*, 9, 220.

Bresenham, J. E., 1965. Algorithm for computer control of a digital plotter. *IBM Systems Journal* 4 (1), 25–30.

Conrad, O., Bechtel, B., Bock, M., Dietrich, H., Fischer, E., Gerlitz, L., Wehberg, J., Wichmann, V., Böhner, J., 2015: System for Automated Geoscientific Analyses (SAGA) v. 2.1.4. *Geoscientific Model Development* 8, 1991–2007.

Gschwandtner, M., Kwitt, R. Uhl. A., 2011. BlenSor: Blender Sensor Simulation Toolbox. *Advances in Visual Computing: 7th International Symposium, (ISVC 2011)* Las Vegas, Nevada, USA, September 26 - 28, 2011.

Kim, S., Min, S., Kim, G., Lee, I. and Jun, C., 2009. Data Simulation of an Airborne LIDAR System. *Proceedings of SPIE-The International Society for Optical Engineering* 7323(1), 1–10.

Kukko, A. and Hyyppa, J., 2009. Small-footprint Laser Scanning Simulator for System Validation, Error Assessment, and Algorithm Development. *Photogrammetric Engineering and Remote Sensing* 75(10), 1177–1189.

Lohani, B. and Mishra, R. K., 2007. Generating lidar data in laboratory: Lidar simulator. *International Archive of Photogrammetry and Remote Sensing* XXXVI(3)W52 of Laser Scanning 2007 and SilviLaser 2007, p. 6.

Lovell, J., Jupp, D., Newnham, G., Coops, N. and Culvenor, D., 2005. Simulation study for finding optimal lidar acquisition parameters for forest height retrieval. *Forest Ecology and Management* 214(1-3), 398–412.

Riegl LMS, 2019. www.riegl.com. Last accessed: 16 January 2019.

UgCS, 2019. www.ugcs.com. Last accessed: 16 January 2019.