GEOMATICS AND ENVIRONMENTAL ENGINEERING • Volume 19 • Number 4 • 2025

https://doi.org/10.7494/geom.2025.19.4.5

Antoni Rzonca[1], Mariusz Twardowski[2]

PyLiGram – Research Application for LiDAR Data Processing Based on Photogrammetric Methods

Abstract: This paper presents the functionality and research possibilities of an application that is based on two concepts: the use of photogrammetric analysis for LiDAR data processing (lidargrammetry), and the assignments of identifiers to cloud points in order to be able to return to the original points after processing without data loss and redundant processing.

The research tool has, thus far, been developed for the implementation of two distinct LiDAR data-enhancement processes. The initial approach involves the altimetric transformation of the LiDAR data (a process that is founded on the principles of stereo model deformation theory), and the second process employs lidargrammetry for the purpose of 3D local point-cloud corrections, global changes, or non-rigid transformation. This is achieved by applying blocks of lidargrams and their subsequent matching and adjustments.

Keywords: LiDAR processing, data integration, lidargrammetry, data enhancement

Received: September 25, 2024; accepted: June 2, 2025

[1] AGH University of Krakow, Faculty of Geo-Data Science, Geodesy, and Environmental Engineering, Department of Photogrammetry, Remote Sensing, and Spatial Engineering, Krakow, Poland, email: arz@agh.edu.pl (corresponding author), https://orcid.org/0000-0003-0333-1000

[2] AGH University of Krakow, Faculty of Geo-Data Science, Geodesy, and Environmental Engineering, Department of Photogrammetry, Remote Sensing, and Spatial Engineering, Krakow, Poland, https://orcid.org/0000-0001-7107-6190

1. Introduction

The improvement of LiDAR data has been a subject of research since this technology was first applied to mapping; the main reason for this is its limited accuracy, and the main factors are the errors in the point coordinates. This paper presents a research tool with the potential to improve the processing of LiDAR data for geometric enhancement; it was created to implement a new approach to LiDAR data processing. This approach includes two basic concepts: photogrammetric analysis for LiDAR processing, and LiDAR point identification by assigned identifiers to restore the original terrain points by the intersections of homologous points (which has never been used in any other known tool).

PyLiGram is a tool for effective and innovative LiDAR data processing based on photogrammetric algorithms and the stored LiDAR 3D point-image relationship, which is the main objective of the presented research.

The following chapter (divided into two parts) presents the state of the art of LiDAR geometric processing. The first part presents the basics of processing, data-evaluation methods, and enhancement methods and gives an overview of the methods that have been implemented in the software that is available on the market. The second part presents enhancement methods based on the integration of LiDAR data with image data. In this section, we present methods that have been proposed by several authors and software that has been designed to process both types of data in order to increase the accuracy of LiDAR data.

In the third chapter, we present our tool; we start by describing the interface and its options and functionalities. Lidargram generation as synthetic-image sampling is the first functionality that is presented and discussed. The process is controlled by several parameters. Two basic problems of the process are (a) empty pixels (pixels without the projection of any LiDAR point) and (b) occlusions (the problem of the geometric filtering of the first plane to project on the lidargram plane). The next functionality that is presented is LiDAR strip adjustment using virtual lidargrams and the stereoscopic model-deformation theory. The last functionality that is described is the use of PyLiGram for extended workflow (also using external software).

The next chapter of the implementation results and discussion presents the factors of lidargram generation: how the used parameters influenced the adjustment results in four tested adjustment algorithms.

After a detailed presentation of the basic features, we outline how we overcame the problems and list the known limitations of the methods that were used. The capabilities of PyLiGram are related to specific research questions of a new approach to LiDAR and photogrammetric data integration. The comparison with mainstream LiDAR-processing software is done within this research and has been published in separate papers.

In the next chapter, we present our research plans that led to the development of our software.

2. State of the Art

2.1. LiDAR Processing

It is a fact that aerial laser scanning (ALS) has been in use for more than 25 years [1]. LiDAR sensors are used in platforms with other sensors such as GPS and IMU [2]. Point-cloud registration is one of the most important and widely researched problems in LiDAR technology [3]. ALS is a mature and developed technology that has been applied worldwide. Practical aspects have been discussed, including different scanning systems, data-processing methods, and software [4]. UAV scanning is becoming increasingly popular; this is mainly due to the advent of lightweight scanners [5].

Many manufacturers offer aerial-mapping platforms that are equipped with scanners. LiDAR is as versatile as photogrammetry and is used for a wide range of applications; these include environmental mapping (such as agriculture [6], forestry [7, 8], and flood analysis [9]) and engineering projects such as dam monitoring [10], building information modeling (BIM) [11], pipelines [12], power lines [13], and architecture [14]. It is also used in archaeology [15].

LiDAR technology has been primarily used as a tool for acquiring digital terrain models (DTMs) and digital surface models (DSMs); as a result, the issue of its geometric quality (such as accuracy, among other factors) has been a key concern and the subject of investigations [16–18]. A number of techniques have been developed to address specific groups of methods that are aimed at improving accuracy, including platform calibration [19–22], trajectory adjustment [23, 24], strip adjustment [25–30], and the application of ground control [31, 32]. Today, LiDAR data is analyzed using machine learning [33, 34] and simultaneous location-and-modeling (SLAM) methods [35].

The evaluation of LiDAR data has been a prominent feature of much research. LiDAR data from UAVs has been the subject of increased investigation and improvement as the technology has matured. The accuracy of LiDAR has traditionally been investigated in two ways: within stripes/between stripes [36, 37], and without trajectory data [38]. As UAV laser scanning (ULS) data gained popularity and its accuracy came under scrutiny, the issue was later investigated in general cases [39] as well as exclusively with direct georeferencing [40, 41] and in the context of specific applications. Bakuła et al. [5] evaluated UAV LiDAR data for the levee monitoring of the Vistula River. Mayr et al. [42] presented two key components of errors in the ULS of grass slopes in the Alps; they assessed registration errors, surface roughness, and positional uncertainties as functions of range, incidence angles, and beam divergence. Fuad et al. [43] evaluated UAV LiDAR as a source of data for DTM generation, and Kucharczyk et al. [44] investigated the accuracy of UAV LiDAR for vegetated terrain.

LiDAR data processing is mostly based on commercial software: the most popular applications are TerraMatch [45] and Riegl's RiProcess [46]. There are many other solutions for data collection, analysis, processing, and visualization, and several reviews have been published [4, 47]. The next generation of LiDAR software is now being developed using artificial intelligence and machine learning [48].

2.2. Data Integration for Enhancement

LiDAR data is increasingly being acquired as parts of combined missions. Photogrammetric data can be adjusted with a high degree of confidence. The idea of LiDAR data enhancement based on the idea of integration seems to be an obvious approach. Several reviews on combined registration methods of imagery and LiDAR data have been published [49–52].

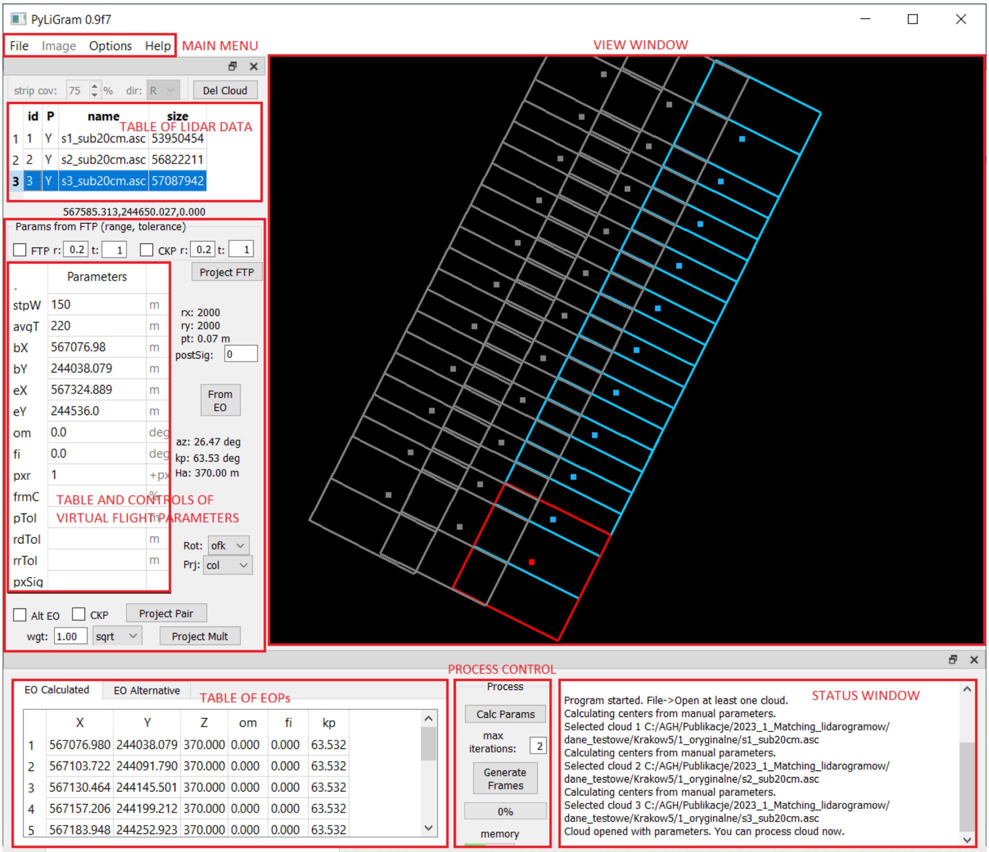
The first data-enhancement process is the calibration of the sensor platform; the trend in research worldwide is to calibrate the platforms without ground-control points or a test field [53, 54]. The calibration of bore sights and eccentrics can significantly increase the accuracy of the data. Another integration option for improving the data is a bundle-block adjustment based on ground control from LiDAR data [55, 56]. The geometric integration of data (co-orientation) can also be done by analyzing image and LiDAR features: linear features [57], 2D/3D point and line correspondences [58, 59], center and corner points of the extracted buildings [60], and other spatial constraints – with additional solutions such as Gabor structure features [61] or the line-point similarity invariant and extended collinearity equation [62]. In general, there are several successful methods that use dense image matching and, further, use a photogrammetric point cloud to integrate it with LiDAR data to improve its accuracy [63] or perform enhancement analysis based on such a hybrid data set [64]. The iterative closest-point algorithm is a typical solution for the geometric adjustment of two sources of data [65]. There are several examples of using depth maps for the co-registration of hybrid data as a further use of dense matching results: depth map stereo [66], or depth maps for point-and-line-feature searches [67]. Point clouds from dense matching and LiDAR can also be refined by quality-based registration [68]. Methods that optimize trajectory data as base LiDAR data are widely represented; these are usually non-rigid methods [69]. The 3D-2D correspondences are used to improve the trajectory [70–73] or for the simultaneous data adjustments of all platform sensors [29]. Pöppl et al. [74] presented a comprehensive review of all of the contemporary trajectory-estimation methods.

Our research focuses on 3D LiDAR adjustment based on a lidargrammetric approach. Lidargrammetry was originally applied as a photogrammetric approach to LiDAR data for stereoscopy and the manual measurements of point clouds [75–77]. These images were sampled from the point clouds as central projection images and used for stereoscopy. Furthermore, synthetic images of the point clouds (lidargrams) and the theory of the vertical deformation of the stereoscopic model can be applied to LiDAR strip adjustment [78], data orientation [79, 80], and feature extraction [81–83].

3. PyLiGram – Research and Processing Tool

3.1. General Layout and Interface

PyLiGram is a tool that was designed for LiDAR data-enhancement research (Fig. 1); this is an open-source program.



strip cov: 75 % dir: R

	id	P	name	size
1	1	Y	s1_sub20cm.asc	53950454
2	2	Y	s2_sub20cm.asc	56822211
3	3	Y	s3_sub20cm.asc	57087942

567585.313,244650.027,0.000

Params from FTP (range, tolerance)

□ FTP r: 0.2 t: 1 □ CKP r: 0.2 t: 1

.	Parameters	
stpW	150	m
avgT	220	m
bX	567076.98	m
bY	244038.079	m
eX	567324.889	m
eY	244536.0	m
om	0.0	deg
fi	0.0	deg
pxr	1	+px
frmC		%
pTol		m
rdTol		m
rrTol		m
pxSig		

rx: 2000
ry: 2000
pt: 0.07 m
postSig: 0

az: 26.47 deg
kp: 63.53 deg
Ha: 370.00 m

Rot: ofk
Prj: col

□ Alt EO □ CKP
wgt: 1.00 sqrt

	X	Y	Z	om	fi	kp
1	567076.980	244038.079	370.000	0.000	0.000	63.532
2	567103.722	244091.790	370.000	0.000	0.000	63.532
3	567130.464	244145.501	370.000	0.000	0.000	63.532
4	567157.206	244199.212	370.000	0.000	0.000	63.532
5	567183.948	244252.923	370.000	0.000	0.000	63.532

max iterations: 2
0%

```
Program started. File->Open at least one cloud.
Calculating centers from manual parameters.
Selected cloud 1 C:/AGH/Publikacje/2023_1_Matching_lidarogramow/
dane_testowe/Krakow5/1_oryginalne/s1_sub20cm.asc
Calculating centers from manual parameters.
Selected cloud 2 C:/AGH/Publikacje/2023_1_Matching_lidarogramow/
dane_testowe/Krakow5/1_oryginalne/s2_sub20cm.asc
Calculating centers from manual parameters.
Selected cloud 3 C:/AGH/Publikacje/2023_1_Matching_lidarogramow/
dane_testowe/Krakow5/1_oryginalne/s3_sub20cm.asc
Cloud opened with parameters. You can process cloud now.
```

Fig. 1. General layout of PyLiGram GUI

The latest developmental beta version of the program for Windows 10 and Ubuntu 22.04 can be downloaded via the following link: https://fotogrametria.agh.edu.pl/~misiek/pyligram/.

The graphical user interface is divided into the following features: main menu, view window, table of LiDAR data to be processed, table and controls of virtual flight parameters (VFP), two-tab table of EOPs of lidargrams, status window, and process-control panel.

The main menu is used to open point clouds in the ASCII format (*.asc), open ground-control text files, and set the working directory (File menu). The Options menu allows one to turn raster generation on or off, among other things.

The Viewer window has no interactive options; it shows the LiDAR program footprints, ground-control points (GCPs), and the cloud during and after processing (Options → Show cloud).

The virtual flight Table of Parameters (ToP) is the most important part of the interface. The main table contains the interior orientation parameters (IOPs) of the virtual camera and the exterior data for calculating the exterior orientation parameters (EOPs) of the lidargrams. All of the data that is associated with each point cloud is automatically stored in a JSON file. Additional controls and buttons are required to process the data within any of the workflows; their functions are explained later. The EOPs table has two tabs: the first (EO calculated) contains the EOPs that have been calculated from the data in the ToP, while the second tab contains those EOPs that can be imported from other software.

There are two basic workflows that can be implemented in PyLiGram; both are dedicated to enhancing LiDAR data using photogrammetric methods and the concept of unique LiDAR point identifiers (ULPIs), which identify an original LiDAR point and its projections on virtual-image planes. The first workflow involves the altimetric deformation of a LiDAR strip based on the theory of stereoscopic model deformation. The second uses lidargrammetry and external lidargram bundle adjustment for the spatial non-rigid adjustment of LiDAR blocks or strips.

3.2. Challenges of Lidargram Generation

As mentioned above, the genesis of lidargrammetry was to create an analogue method of measuring LiDAR data by the stereo viewing of images instead of point clouds in the top/front/right 2D viewers of CAD applications. PyLiGram is actually dedicated to LiDAR data enhancement, but the basis is the generation of stereo image pairs of predefined IOPs and EOPs.

In addition, the raster-generation process is parallel to the assignments of IDs to a point. Each point cloud has a unique LiDAR point identifier (ULPI); these ULPIs allow the process of generating lidargrams to be reversed and a new point cloud to be generated after analytical operations have been applied to the lidargrams.

After importing the point cloud into PyLiGram, all of the parameters have to be set manually, or they can be read from a previously generated JSON file with the same name as the point cloud. First, the focal length, pixel size, and image format are defined; after entering the ground coordinates of the start and end points, image overlap, strip width, and average terrain height, the EOPs of the lidargrams are calculated and inserted into the EO Calculated tab of the EOPs. Another way to use EOPs is via a text file; EOP files can be imported using the From EO button. Any change to the data in the Virtual Flight Parameters table results in an update of the JSON file. The IOPs and EOPs are used, and the maximum size of the lidargram format is used. The process is started by pressing the Generate Frames button. The fiducial coordinates are calculated from the ground coordinates, and each point is assigned an ULPI. These fiducial coordinates (in millimeters) with ULPIs are used in further processing to compute a new point cloud within the application of the stereo model deformation theory or non-rigid transformation of the point cloud. Such a unique approach (instead of pixel coordinates) ensures that the accuracy of the original points is not lost. Optional raster generation is also possible if this is required for the lidargram-matching process within the second basic research workflow (described in the following section).

Synthetic-image generation from point clouds faces two problems: the problem of empty pixels, and the problem of occluded areas.

Empty Pixels

The first problem is empty pixels where there is no projected LiDAR point cloud. Initially, there are many empty pixels between the pixels of any projected LiDAR point coordinates (Fig. 2a). This can be solved in two ways and can be successful until the blurring of the image degrades its quality: enlarging the virtual sensor pixel (Fig. 2b), or increasing the number of lidargram pixels that are colored around the central pixel (Fig. 2c).

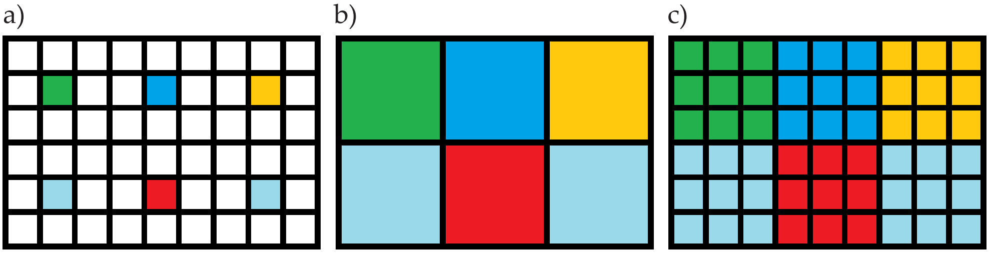


Fig. 2. Simulated results of point-cloud projection on lidargram: a) simple projection with empty pixels; b) magnified virtual-camera pixel size; c) pixel-range option with parameter = 1 px

In fact, both ways can give the same effect; however, we prefer the second way – without changing any parameters of the virtual sensor (like pixel size) for the sake of future research (among others, the integration of photo and LiDAR data by co-matching two sets of images with the same IOPs). To solve this problem, we use the so-called “pixel-range” parameter (pxr – abbreviation in the PyLiGram GUI), which defines the number of pixels around which to color. If this parameter is equal to 0 (Fig. 2a), only the central pixel is colored; if it is equal to 1, an additional pixel is colored in all directions. The total number of colored pixels is nine – one central pixel, and the eight pixels around it (Fig. 2c). For pxr = 2, the total number of colored pixels is 25 (1 + 8 + 16).

Another option is implemented during the process: Gaussian blurring. The blurring during lidargram generation is set by the so-called "pixel-sigma" parameter (pxSig – shortcut in the PyLiGram GUI). This represents the standard deviation of the color in the pixels starting from the central pixel. For example, with settings of pxr = 2 and pxSig = 2, the central pixel takes on the color of the original point; the color of the surrounding pixels decreases according to the Gaussian function (with a standard deviation of 2 px). Pixel-sigma parameter only works within the number of pixels that are defined by the pixel-range parameter. For a better understanding, Figure 3 shows a comparison of three sample images of a single red LiDAR point projection.

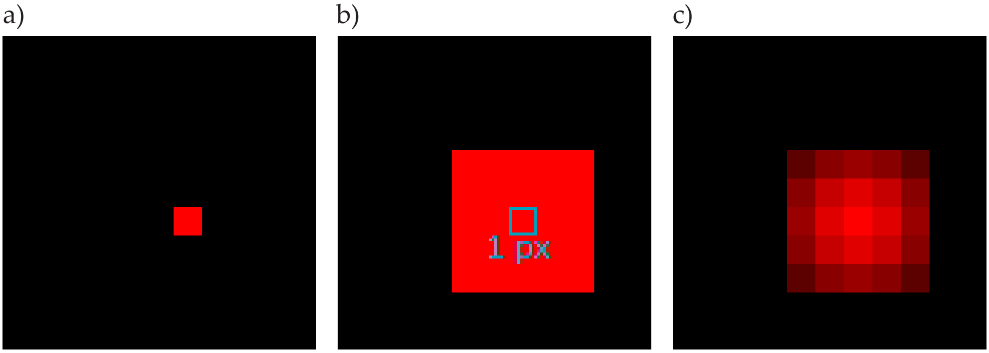


Fig. 3. Simulated results of projection of single red point:
a) simple projection with pxr = 0 and pxSig = 0; b) pxr = 2 and pxSig = 0;
c) pxr = 2 and pxSig = 2

Occlusions

The second problem that needs to be solved when generating lidargrams is occlusion. Some areas that are scanned by LiDAR cannot be projected onto a lidargram, as the point cloud is continuously acquired and lidargrams are only acquired from certain points in space. In addition, the original trajectory of the LiDAR is unknown and may not be exactly the same as the virtual flight line that is defined by VFP. To solve this problem, two parameters were implemented (Fig. 4).

The first is rdTol – a height tolerance for points that are projected into a pixel; all are compared to the highest point and are discarded if they are lower than rdTol. Another parameter – rrTol – defines the search radius if there are higher points in the neighborhood as defined by rrTol.

The result of using these parameters is shown in Figure 5; it shows the roof of a building. Looking from this direction, the wall of the building and part of the road are occluded. The first example shows a lidargram that is generated with pxr = 1 and rdTol and rrTol = 1 (Fig. 5a). There are many empty pixels, but the occluded areas (such as the wall of the building and the road) are not projected onto the lidargram image. The second sample of the same pixel range was generated without filtering, and there are pixels on the roof (inside the red ellipse) that represent the original LiDAR points of the occluded building wall and road (Fig. 5b). The same result can be seen in the next two samples (with and without filtering and generated with pxr = 2). The third sample is correctly generated and shows that the filter works correctly (Fig. 5c). Like the second, the fourth sample contains pixels in the roof area of the occluded parts (Fig. 5d).

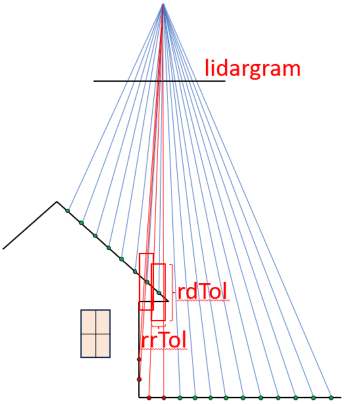


Fig. 4. Point-cloud filtering during projection of occluded areas – green points are taken into account to generate lidargram; red ones are not

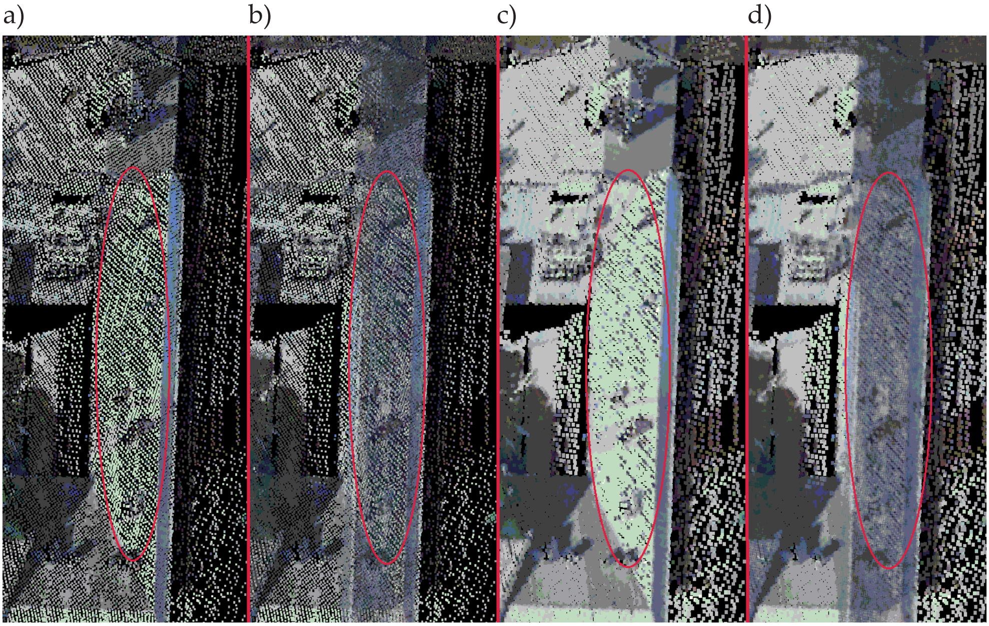


Fig. 5. Four samples of lidargram presenting roof and occlusion:
a) pxr = 1 (filter on); b) pxr = 1 (filter off); c) pxr = 2 (filter on);
d) pxr = 2 (filter off)

Despite the use of these parameters, the quality of the lidargrams is limited; more-sophisticated methods such as post-processing Gaussian blurring and color interpolation will be explored in the near future.

3.3. Model-Deformation Functionalities

The first LiDAR data-enhancement method that is implemented in PyLiGram was described in [78]. The deformation theory of the stereoscopic model was applied to the deformation of a non-rigid LiDAR strip; this deformation is defined by the vertical discrepancies of four corners of the strip. The values of the discrepancies should be defined by ground-control points (GCP) or patches. The flexibility of the method allows each of the corners to be moved up or down by a different vector (Fig. 6).

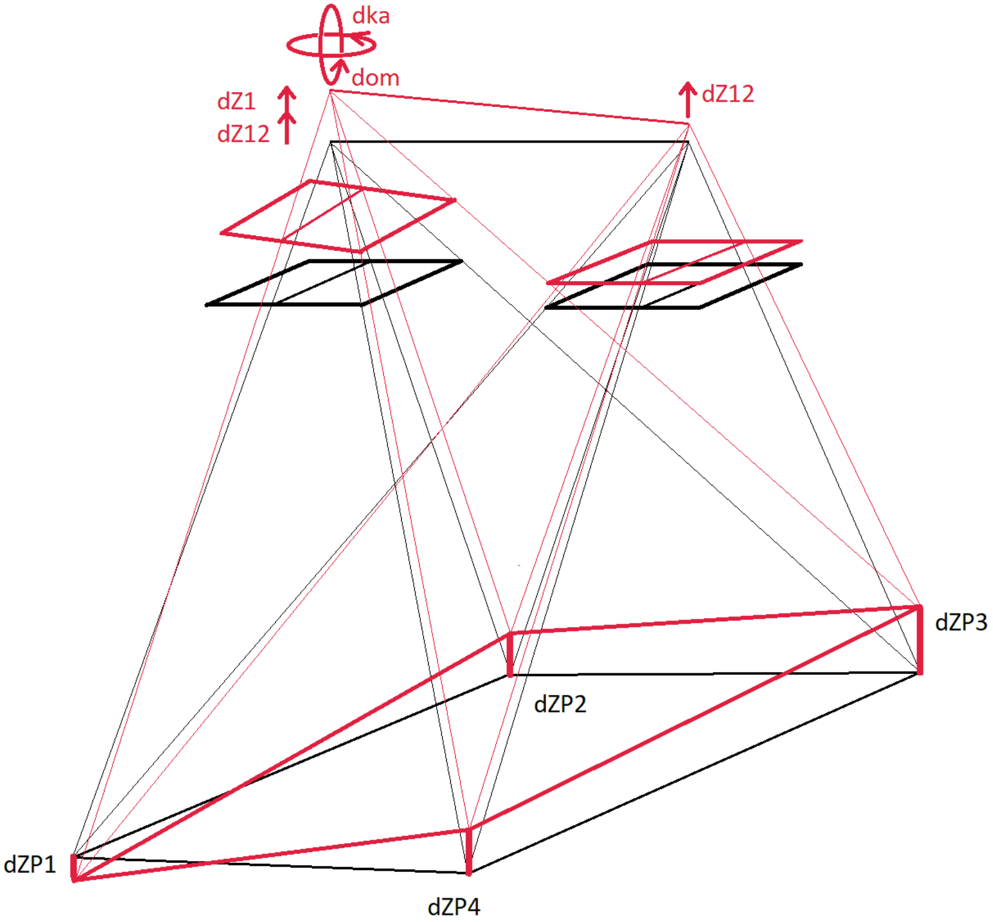


Fig. 6. Strip-height deformation by changing lidargrams' EOPs

The workflow starts by opening the point cloud to be processed; the virtual camera parameters must be inserted into the virtual flight parameters table, and the text file containing the GCPs must be opened. Parameters *r* (range) and *t* (tolerance) define the radius and height tolerance, respectively, of the vertical cylinder that is defined for each of the GCPs. The symmetry axis of the cylinder is vertical and passes through the GCP. The heights of the points of the processed cloud within the cylinder are averaged, and the height difference is calculated with respect to the GCP. Once the GCPs are selected, the EOPs of the lidargrams are calculated. The *X* and *Y* coordinates of the first projection center are in the middle of the first two GCPs, while the *X* and *Y* coordinates of the second projection center are in the middle of the next two GCPs. Both the IOPs and EOPs are automatically saved in a JSON file with the same name as the point cloud. Based on these four differences, the algorithm calculates four single steps according to the possible changes of the relative orientation parameters (ROPs) of the virtual lidargram model: the height difference of both projection centers dZ_{12}, the height difference of the first projection center (dB_Z), the difference of the omega rotation of the first lidargram (*dom*), and the difference of the kappa rotation of the first lidargram (*dka*) according to Equation (1):

$$dZ = dZ_{12} - \frac{(X-B)}{B}dB_Z + \frac{XY}{B}dom - \frac{YH}{B}dka \qquad (1)$$

When the new ROPs are applied, the new point cloud is forward-intersected.

The method gives perfectly correct effects – provided that the GCPs are in a regular rectangle; if not, the curvature of the edge of the processed point cloud (the edge that ends the strip that is perpendicular to the flight line) may appear. When the next part of the strip is processed, the curvature will appear on the common edge in the opposite direction; this is one reason for the discrepancy. The next disadvantage is that the point cloud can only be corrected vertically. We have not yet implemented another possible option to change the *X* and *Y* of the horizontal shift of the point cloud. Changing the azimuth of the baseline (connecting the two centers of the projection of the lidargrams) would allow for the global kappa rotation of the point cloud. This was not implemented because it was decided to develop the second much more flexible and universal method: spatial point-cloud enhancement based on GCPs, and lidargram-matching.

The method was tested on synthetic, semisynthetic, and real data, as was its implementation in PyLiGram [84]. The GCPs were defined as selected points of the original cloud, and their heights was changed to simulate vertical discrepancies (they were not measured in the line of sight – these can be referred to as virtual GCPs [VGCPs]). Such an approach is useful for testing the method and its implementation, as the coordinates of the VGCPs are virtually marked and can be measured without error. It is a known problem that the nature of LiDAR data makes it difficult to unambiguously mark, identify, and measure VGCPs – especially on the horizontal plane. This problem is avoided with VGCPs, which do not need to be measured on the point cloud. This is only useful for method and software testing.

3.4. Matching of Lidargrams

Another application of lidargrammetry to LiDAR data processing is a one-step enhancement method that is based on matching lidargrams and their block adjustments using GCPs for flexible point-cloud transformation. This method is the subject of a separate paper (which is currently under review).

The method uses external matching and adjustments by Agisoft Metashape. The workflow starts from PyLiGram using the externally adjusted EO calculation and then processes the point-cloud intersection after returning to PyLiGram (Fig. 7).

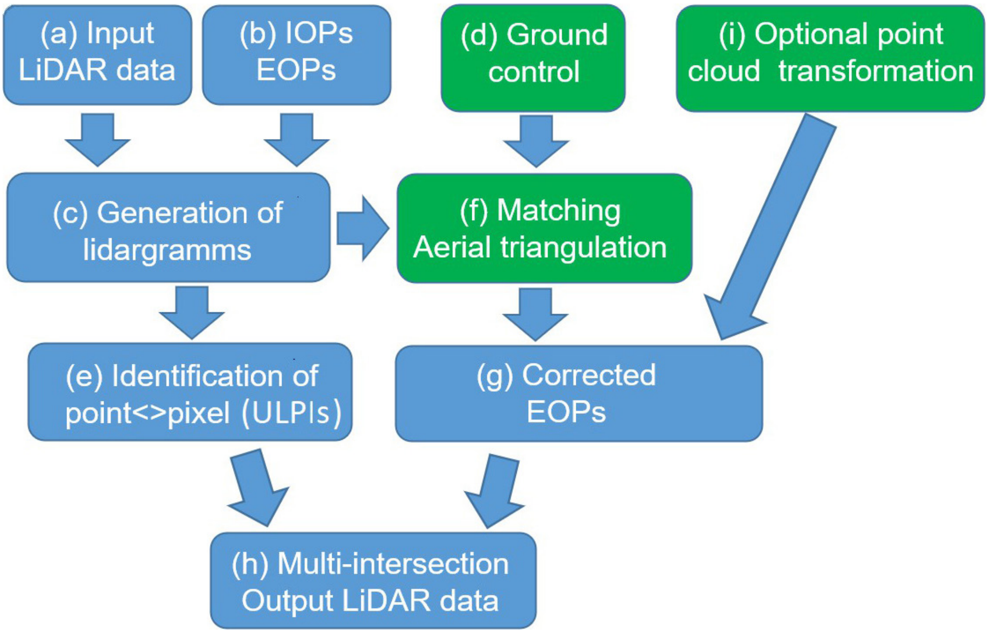

(a) Input LiDAR data
(b) IOPs EOPs
(d) Ground control
(i) Optional point cloud transformation
(c) Generation of lidargramms
(f) Matching Aerial triangulation
(e) Identification of point<>pixel (ULPIs)
(g) Corrected EOPs
(h) Multi-intersection Output LiDAR data

Fig. 7. Workflow of lidargram-matching for flexible 3D point-cloud transformation and registration

The same method of VGCPs was used to test this PyLiGram functionality. The text file that contains the VGCPs is selected after importing the point clouds; the XML file of their virtual measurements on the lidargram is generated at the same time when the lidargram is generated. The XML contains the data in the format that is required by Agisoft. The rasters must be saved to a disc for matching and adjustments in the external software. The adjustments use a different set of coordinates to the GCPs, as they are used as block-deformation information.

The matching and strip adjustment use the same camera as defined during the generation of the lidargram. The a priori accuracies of the measurement errors, input-data errors, and approximated EOPs are adapted to the need of the flexibility of the block: the low accuracy of the measurements of the tie points is defined. The accuracies of the coordinates of the GCPs and the measurement errors of the GCPs are relatively high in order to deform the block according to the locations of the GCPs and to allow the block of lidargrams to be deformed during the block adjustment. The resulting EOPs are imported into PyLiGram after the adjustments, and all of the points of the deformed point cloud are recalculated by least-squares adjustment as a multiple intersection using these new EOPs and according to the ULPIs.

Several specific functionalities have been implemented in PyLiGram to make it able to flexibly deform a point cloud, but the idea is based on the use of ULPIs and lidargram-matching.

4. Implementation Results and Discussion

The results of all three options for solving the black-pixel problem were tested by matching lidargrams. Four procedures were repeated for three test areas: Krakow’s city center, Loosdorf (Austria), and Bieruń (a village in Poland). The first procedure was a standard COLMAP software reconstruction [85], while the next three were combinations of 3DOM matching pipelines based on deep learning [86] and COLMAP reconstruction, adjustment, and accuracy reporting. The pipelines were as follows: a combination of the Superpoint feature search method [87] and Superglue matching [88] (Superpoint+Superglue); a Lightglue feature search [89] and Superglue matching; and the DISK method [90]. This was repeated for four levels of the matching quality of the process (lowest/low/medium/high). In general, the quality was defined by the ratio of the image pixel that was used for the matching to the original pixel. For the high quality, the ratio of the original pixel to the matching pixel was 1:1; this was omitted because it was too time-consuming and resource-intensive in our preliminary tests. The low-quality level was chosen for our comparison, as it was common to all of the test data. Such a level reduced the original resolution by a factor of 4 for the processing.

The testing of the several variants of pixel-range and pixel-sigma parameters for Colmap and the Superpoint+Superglue, Superpoint+Lightglue, and Disk+Lightglue pipelines was controlled by two parameters: the number of matches, and the reprojection error. In order to summarize the results, the sum of the matches and the mean reprojection error values were calculated for each pipeline across each of the pixel-range and pixel-sigma parameters.

Three test fields were selected to test the lidargram-generation process: Krakow’s city center, Loosdorf, and Bieruń. Table 1 shows the basic technical data from these data fields.

Table 1. Test-field data

Specification	Krakow's city center	Loosdorf	Bieruń
Average point distance (APD) [cm]	15	3	10
Source	blocks of data	strip	strip
ALS/ULS	ALS	ALS	ULS

4.1. Krakow Test Field

The LiDAR data for Krakow's city center came from the ISOK project (Informatyczny System Ochrony Kraju [Computerized National Protection System]). The data was used in blocks rather than strips; this approach made it possible to access dense LiDAR data from a highly urbanized area. The intensity point cloud had an average point distance (APD) of about 15 cm and was acquired with a red-light scanner. Stereo images of the lidargrams were generated using virtual large-format aerial camera EOPs; these were predefined to be equivalent to real photogrammetric data. The ground-sampling distance of the lidargrams was about 10 cm, and several configurations of the pixel-range and pixel-sigma parameters were used.

The number of matches and the reprojection error by relative orientation are shown in Tables 2 and 3, respectively. Twenty variants of the pixel-range and pixel-sigma parameter settings were calculated, along with four matching methods and four quality levels.

Table 2. Numbers of matches for low-quality variants of Krakow's city center test data (maximum values for pipeline/pixel range in bold)

Test ID	Pixel range [px]	Pixel sigma [px]	Number of matches			
			Colmap	3DOM Disk+Lightglue	3DOM Superpoint+Lightglue	3DOM Superpoint+Superglue
1	1	0	**3854**	1051	2338	2232
2	1	1	3307	**1057**	**2368**	**2394**
3	2	0	**3250**	**1083**	**2979**	**2977**
4	2	1	2845	953	2531	2460
5	2	2	3126	1076	2889	2870
6	3	0	**3430**	1137	**3894**	**3910**
7	3	1	397	971	2459	2458
8	3	2	2865	1111	3121	3118
9	3	3	3186	**1149**	3375	3372
10	4	0	**3649**	**1217**	**4162**	**4164**
11	4	1	56	1031	2239	2266
12	4	2	3082	1105	3053	3044

Table 2. cont.

13	4	3	3352	1203	3350	3336
14	4	4	3274	1210	3565	3560
15	5	0	3737	**1281**	**4012**	**4001**
16	5	1	25	978	2059	2133
17	5	2	**3796**	1132	2958	2994
18	5	3	3006	1198	3278	3289
19	5	4	3360	1217	3445	3458
20	5	5	454	1262	3647	3644
Sum			54,051	22,422	61,722	61,680
Std. dev. of best (bold)			243	93	776	767

After analyzing the results of Table 2, we can make the following assumptions:

1. Colmap matched the lidargrams with the highest numbers of points; however, there were some exceptions.
2. The pipeline Disk+Lightglue gave the lowest number of matches, while the Superpoint+Lightglue and Superpoint+Superglue pipelines provided similar results.
3. The best results could be found for pxSig = 0.4; as an exception, the lidargrams with smaller pixel ranges than the photo pixel sizes matched better when pixel sigma was on.
4. The highest number of matches could be found for the Superpoint+Lightglue variant.
5. The experiment of the Gaussian blurring of the pixels within the pixel range gave negative results.
6. Comparing the results of Test IDs 1 and 2, there were more matching points after enlarging the pixel areas of the lidargrams.
7. It is also worth mentioning that the maximum total sum of matches for each method could be found in the Superpoint+Lightglue method; the most-similar results through all of the methods could be found for the Disk pipeline.
8. The results of Colmap were the largest.
9. If we consider only the maximum results for each pipeline and pixel size case (bold), we can observe that the mean scatter of the results could be found for the Colmap pipeline, the smallest scatter – for the Disk pipeline, and the largest – for both Superpoint methods.

Figure 8 shows the distribution and the number of matches. As mentioned above, the highest number of matches could be found in the Colmap results (Fig. 8a), the lowest number – in the Disk+Lightglue results (Fig. 8d); the Superpoint methods were the best of the deep-learning methods (Fig. 8b, c). The distributions of the matches were correct and relatively uniform for all four methods.

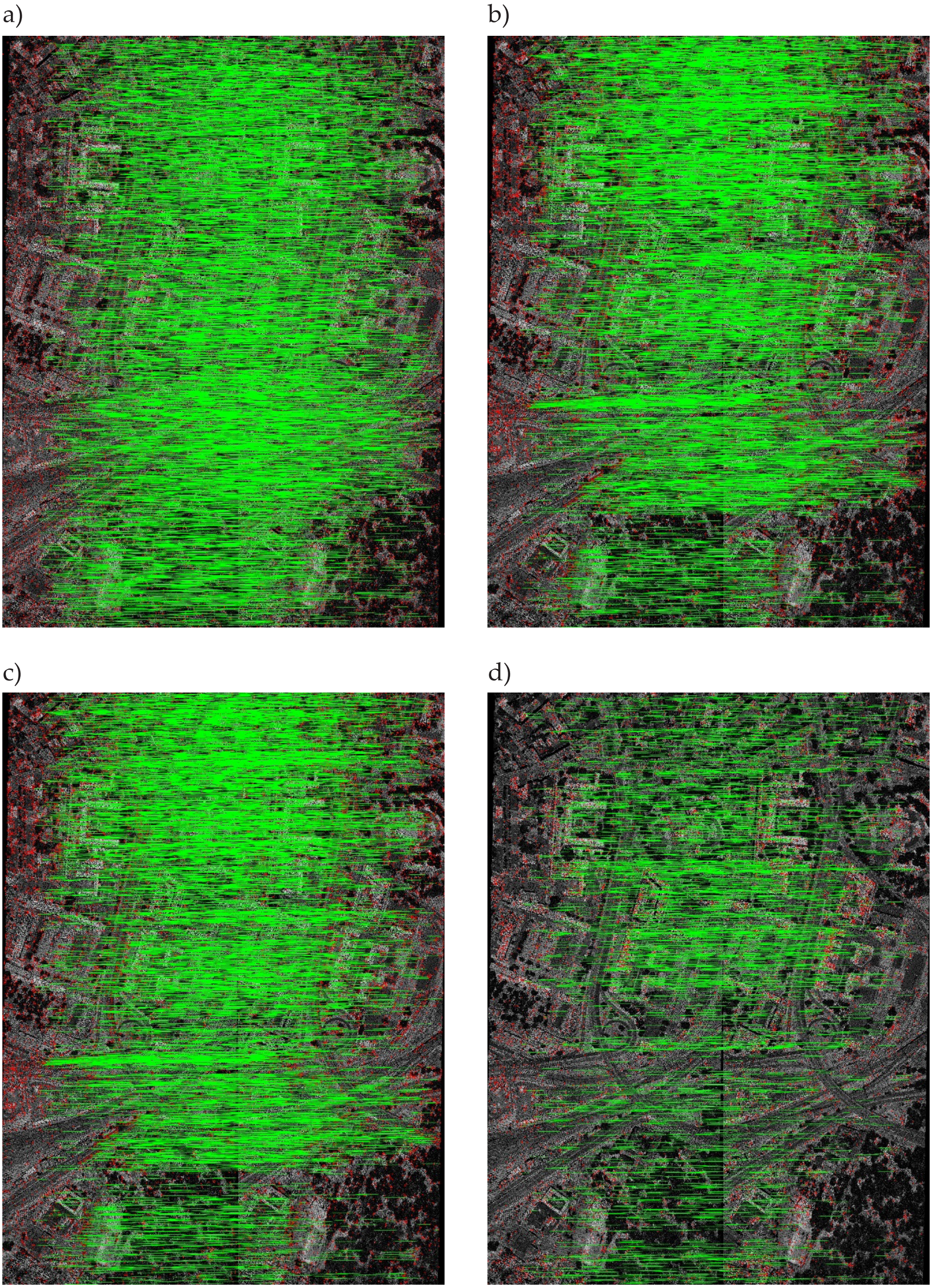


Fig. 8. Distribution of matches of four compared feature-matching algorithms: a) SIFT; b) Superpoint+Superglue; c) Superpoint+Lightglue; d) Disk+Lightglue

Table 3. Reprojection errors for low-quality variants of Krakow's city center test data (minimum values for pipeline/pixel range in bold)

Test ID	Pixel range [px]	Pixel sigma [px]	Reprojection errors [px]			
			Colmap	3DOM Disk+Lightglue	3DOM Superpoint+Lightglue	3DOM Superpoint+Superglue
1	1	0	1.071	**1.275**	1.645	**1.606**
2	1	1	**1.022**	1.302	**1.631**	1.633
3	2	0	1.084	1.362	**1.578**	1.586
4	2	1	**0.999**	**1.341**	1.629	1.624
5	2	2	1.101	1.345	1.587	**1.569**
6	3	0	1.028	**1.275**	**1.556**	**1.559**
7	3	1	**0.966**	1.375	1.704	1.691
8	3	2	1.146	1.331	1.607	1.595
9	3	3	1.025	1.298	1.558	1.570
10	4	0	1.016	**1.291**	**1.553**	**1.559**
11	4	1	**0.706**	1.377	1.757	1.733
12	4	2	1.104	1.295	1.621	1.607
13	4	3	1.042	1.290	1.565	1.569
14	4	4	1.087	1.287	1.547	1.546
15	5	0	0.962	1.312	**1.482**	**1.482**
16	5	1	**0.391**	1.379	1.629	1.751
17	5	2	0.928	1.294	1.613	1.623
18	5	3	1.057	**1.287**	1.546	1.548
19	5	4	1.018	1.290	1.538	1.548
20	5	5	1.335	1.295	1.539	1.540
Mean			0.987	1.316	1.597	1.600
Std. dev. of selection (bold)			0.270	0.027	0.054	0.047

After analyzing the projection errors in Table 3, we can make the following assumptions:

1. The Colmap pipeline provided the best results – even 0.391 px for a five-pixel area and one pixel-sigma case.
2. The Disk-reprojection errors were higher than those of Colmap but lower than those of the two Superpoint pipelines.
3. It was significant that the blurring by the pixel-sigma parameter gave better results for the Disk pipeline but not for the other methods.
4. The dispersion of all of the results was highest for Colmap – even when only the best results were taken into account.
5. The results of the Superpoint methods were the worst.

Comparing the numbers of matches and the numbers of reprojection errors, the Colmap pipeline generally provided the best results in both cases; the general scatter and the scatter of the best results were medium for both of the Superpoint methods. The scatter of the reprojection error of Disk was generally the smallest, while the largest could be found in the Colmap pipelines. This meant that Colmap was the most sensitive to changes in the pixel range and pixel sigma. All of the deep-learning methods were ten-times-less sensitive.

4.2. Loosdorf Test Field

The Loosdorf data was acquired using a Riegl VQ-1560i imaging system. The IR band scanner data was used to generate the lidargram, the virtual camera IOPs were equal to the PhaseOne iMX-100 camera IOPs, and the EOPs were applied from real adjusted photogrammetric data. The GSD of the photos was approximately 7.4 cm, and the APD of the LiDAR points was 3 cm. The results of the four pipelines that were analogous to Krakow's city center data set are presented in Table 4 (the number of matches) and Table 5 (the reprojection error).

Table 4. Numbers of matches for low-quality variants of Loosdorf test data (maximum values for pipeline/pixel range in bold)

Test ID	Pixel range [px]	Pixel sigma [px]	Number of matches			
			Colmap	3DOM Disk+Lightglue	3DOM Superpoint+Lightglue	3DOM Superpoint+Superglue
1	1	0	**2259**	**1005**	**1444**	**1442**
2	1	1	1822	960	1385	1378
3	2	0	**2788**	**928**	**2986**	**3027**
4	2	1	2017	754	1764	1757
5	2	2	2289	897	2680	2728
6	3	0	**3029**	**900**	2479	2506
7	3	1	1872	810	1939	1917
8	3	2	2531	878	2745	2793
9	3	3	2463	882	**2771**	**2801**
10	4	0	**3227**	**942**	2321	2331
11	4	1	1833	873	1807	1791
12	4	2	2129	914	2320	2314
13	4	3	2403	920	2634	2660
14	4	4	2600	934	2663	**2692**
Sum			33,262	4678	31,938	32,137
Std. dev. of best (bold)			418.3	45.4	690.0	659.6

To resume the results from Loosdorf in the set of games, we observed the following:

1. Colmap found the highest number of points – 3029 was the maximum number for pxr = 3 and pxSig = 0.2; Disk reprojection showed the worst effectiveness, and both Superpoint methods gave similar results not (significantly worse than the Colmap method).
2. Applying blur by using the pixel-sigma parameter was not helpful; in some cases, however, it gave slightly better results.
3. The scatter of the number of matches was the highest for the two Superpoint methods and the lowest for the Disk. The scatter of the reproduction errors of the lowest errors was the lowest for the Superpoint+Lightglue method and the highest for Colmap.
4. This meant that the Colmap results were much more dependent on the pixel-range and pixel-sigma parameters than the deep-matching methods.

Table 5. Reprojection errors for low-quality variants of Loosdorf test data (minimum values for pipeline/pixel range in bold)

Test ID	Pixel range [px]	Pixel sigma [px]	Reprojection error [px]			
			Colmap	3DOM Disk+Lightglue	3DOM Superpoint+Lightglue	3DOM Superpoint+Superglue
1	1	0	0.989	**1.294**	1.520	1.528
2	1	1	**0.942**	1.336	**1.468**	**1.455**
3	2	0	**0.788**	**1.225**	**1.299**	**1.298**
4	2	1	0.981	1.276	1.458	1.461
5	2	2	0.923	1.198	1.361	1.376
6	3	0	**0.749**	1.169	1.277	**1.284**
7	3	1	0.930	1.360	1.433	1.446
8	3	2	0.868	**1.120**	**1.314**	1.333
9	3	3	0.857	1.121	1.317	1.324
10	4	0	**0.735**	**1.191**	**1.250**	**1.259**
11	4	1	0.914	1.318	1.425	1.436
12	4	2	0.864	1.193	1.349	1.343
13	4	3	0.855	1.207	1.311	1.325
14	4	4	0.813	1.180	1.291	1.301
Mean			0.872	1.228	1.362	1.369
Std. dev. of selection (bold)			0.118	0.055	0.094	0.020

The reprojection-error analysis can be concluded as follows:

1. The highest accuracy was obtained with Colmap; the results of the other methods were slightly worse.
2. The best results of the deep-learning methods were obtained by Disk.
3. Using the pixel-sigma parameters provided better results in some cases, but this was not a general rule.
4. Colmap had the largest scatter of results, while the results from Superpoint-Superlight were the most similar.

Comparing the numbers of matches and the numbers of reprojection errors, the numbers of points generally indicated that the Superpoint methods were the most effective; however, the best accuracy of matching was the Colmap method. The scatter of the results was higher for all of the pixel-sigma variants. When selecting the best, the scatter of the results was generally lower for the deep-matching methods.

4.3. Bieruń Test Field

The third data set was a UAV data set from a DJI Phantom Pro RTK drone with a Riegl VUX-1UAV scanner. LiDAR programs were generated using IOPs from the Phantom Pro's 88 mm camera. The point cloud had about 10 cm APD, while the GSD of the photos had 1.3 cm. The results of the matching according to the LiDAR program-generation parameters are shown in Tables 6 and 7.

Table 6. Numbers of matches for low-quality variants of Bieruń test data (maximum values for pipeline/pixel range in bold)

Test ID	Pixel range [px]	Pixel sigma [px]	Number of matches			
			Colmap	3DOM Disk+Lightglue	3DOM Superpoint+Lightglue	3DOM Superpoint+Superglue
1	1	0	**2792**	1834	**866**	**904**
2	1	1	2410	**1839**	742	742
3	2	0	1991	**2018**	**896**	**894**
4	2	1	2179	1797	497	518
5	2	2	**3417**	1989	872	868
6	3	0	2989	**1988**	790	798
7	3	1	**4410**	1761	395	396
8	3	2	1565	1968	754	745
9	3	3	2244	1975	**887**	**895**
10	4	0	2826	2031	631	525
11	4	1	**5682**	1690	305	304

Table 6. cont.

12	4	2	1667	1959	567	548
13	4	3	606	**2053**	711	706
14	4	4	1752	2033	**752**	**759**
Sum			36,530	26,935	9665	9602
Std. dev. of best (bold)			1271.8	116.2	187.6	195.3

Bieruń's results can be summarized as follows:

1. Colmap found the highest number of points – 5682 was the maximum for pxr = 4 and pxSig = 1.2; the Superpoint+Superglue pipeline had the worst efficiency, both of the Superpoint methods gave similarly low results, and the Disk pipeline results were in the middle.
2. Applying image blur using the pixel-sigma parameter was not helpful; in some cases, however, it gave slightly better results.
3. The scatter of the number of matches was highest for the Colmap results, Disk had the lowest number of matches, and both Superpoint methods gave average results.
4. This meant that the Colmap results depended much more on the pixel-range and pixel-sigma parameters than the deep-matching methods.

Table 7. Reprojection errors for low-quality variants of Bieruń test data (minimum values for pipeline/pixel range in bold)

Test ID	Pixel range [px]	Pixel sigma [px]	Reprojection error [px]			
			Colmap	3DOM Disk+Lightglue	3DOM Superpoint+Lightglue	3DOM Superpoint+Superglue
1	1	0	0.415	**0.855**	1.438	1.465
2	1	1	**0.369**	0.892	**1.433**	**1.443**
3	2	0	0.735	0.851	1.485	1.471
4	2	1	**0.387**	0.913	1.462	1.598
5	2	2	0.472	**0.845**	**1.482**	1.474
6	3	0	0.653	0.877	1.375	1.359
7	3	1	**0.265**	0.975	1.530	1.592
8	3	2	0.739	0.876	1.473	1.457
9	3	3	0.691	**0.864**	**1.291**	**1.310**
10	4	0	0.596	0.966	1.511	1.533
11	4	1	**0.248**	1.015	1.682	1.666

Table 7. cont.

Test ID	Pixel range [px]	Pixel sigma [px]	Reprojection error [px]			
			Colmap	3DOM Disk+Lightglue	3DOM Superpoint+Lightglue	3DOM Superpoint+Superglue
12	4	2	0.581	0.940	1.476	1.457
13	4	3	0.831	**0.932**	1.429	1.422
14	4	4	0.687	0.957	**1.390**	**1.379**
Mean			0.548	0.911	1.461	1.473
Std. dev. of selection (bold)			0.188	0.054	0.088	0.097

The results of the reprojection-error analysis were as follows:

1. The highest accuracy was obtained with Colmap; the results of the other methods were worse (but acceptable).
2. Disk gave the best results for the deep-learning methods.
3. Using pixel-sigma parameters gave better results in most cases, but this was not a general rule.
4. Colmap had the largest scatter of results, Superpoint-Superlight had the most similar results, and the Disk pipeline had the most-coherent results.

Analyzing the results in terms of the numbers of matches and reprojection errors, we could generally consider the Colmap method to be better in terms of the number of matches that were found and the final accuracy. It should be noted that the scatter of the number of matches was relatively high; this was due to the different settings of the pixel range and pixel sigma. The scatter of the reprojection error was also high.

4.4. Discussion

The results that were presented for all of the test data proved that the matching of the lidargrams from different original lidars was possible and effective. The use of four pipelines (one standard pipeline from Colmap, and three deep-learning pipelines from 3DOM) allowed for a broader analysis of the problem and ensured that the lidargram-generation procedure of PyLiGram provided synthetic images that could be matched. Two additional parameters were implemented, and the number of combinations of these parameters determined the number of cases that were computed.

The number of matched points was the first parameter that was used to evaluate the used pipelines. For the Krakow and Loosdorf data, where the GSD/average point ratio was similar (approximately 3:2 and 2:1, respectively), the number was high, with Colmap and both Superpoint pipelines giving the best results. In the case of Bieruń, where the ratio was the opposite (about 1:6), Colmap gave the best results.

The spreads of the results were highest with the Colmap pipeline for the Krakow and Bieruń data and with the Superpoint pipeline for Loosdorf. There was no obvious trend.

Similarly, the reprojection errors were the lowest for Colmap for Krakow and Loosdorf (as well as for Bieruń). The worst results in terms of accuracy were obtained by the two Superpoint pipelines; this can be explained by the facts that the models of the deep-learning methods were trained on real photo data and the matching process was much more flexible (taking imperfect matches into account). Colmap's reprojection errors were less coherent than those of the other pipelines.

The next aspect to discuss is whether the pixel-range option supported the matching process. When analyzing all of the results, there was now a clear correlation between the pixel range and the number of matches. In the Krakow test, increasing the pixel range did not change anything in the Colmap pipeline; in the deep-learning pipelines, it slightly supported the process. The original point cloud was very dense and there were not many empty pixels on the lidargram that were generated with a pixel range that was equal to 0. In the Loosdorf data, it helped to start with a pixel range that was equal to 2 in order to fill the empty pixels. In Bieruń, the pixel size of the virtual camera was chosen to fill the gaps of the very sparse point cloud.

An additional experimental parameter was the pixel sigma; the experiment did not give a clear positive result. Most of the obtained numbers of matches and reprojection errors were for a pixel sigma that was equal to 0; there were exceptions, but it was impossible to state a principle.

The last point of discussion is which pipeline provided the most coherent results? This analysis was based on the best results for each group of calculations within the same pipeline and within the same pixel range. The best results were selected, and these were analyzed. The most coherent results in terms of the numbers of matches and reprojection errors were obtained by the Disk+Lightglue pipeline in nearly all of the cases.

5. Conclusions

This paper presents the development and functionalities of the PyLiGram research tool. PyLiGram has been developed for more effective and innovative LiDAR data processing; it is based on photogrammetric algorithms that are applied to LiDAR data through synthetic-image generation and processing. An additional important solution that was applied within PyLiGram's processing was the unique LiDAR point identifier (ULPI) assignment of original LiDAR points and their projections onto lidargrams.

Three processes are described and discussed: lidargram generation, a model-deformation method for flexible LiDAR strip transformation, and adjustments of blocks of LiDAR data by external matching as well as adjustments of lidargrams for multi-intersection.

Several innovative solutions were applied for the correct generations of lidargrams. The first solution was to enable the generation of a synthetic image that looked like a real camera image; this was important for manual stereoscopic observation, which was the first goal of lidargrammetry. Several tests and experiments were carried out to improve the quality of the generated images. Two main problems were solved: how to fill empty pixels, and how to avoid projecting points into the occluded areas.

The algorithm for filling the empty pixels was tested by the quality of the match, which was measured by the number of matches that were found and the reprojection error. The first parameter was the pixel range; it improved the results until the APD was similar to the GSD of the lidargram. However, it was generally more useful for improving direct observations and image perceptions than for matching. Four pipelines were tested, and most of the results indicated that Colmap was the most effective method. The use of the pixel-range parameter was not confirmed to be critical for success in matching.

Another parameter was used: pixel sigma for Gaussian blurring. The results of the number of matches and the reprojection error were not improved by this parameter.

The overall conclusion was that the applications of these parameters (pixel range and pixel sigma) had little influence on the matching results. Despite these results, the lidargram-generation algorithm facilitated the matching of these images.

The second research functionality of PyLiGram is the application of model-deformation theory for the flexible height transformation of LiDAR strips. This innovative photogrammetric approach is supported by separately described research [78]. The method has some limitations; however, it general allows for the creation of a LiDAR strip that is based on ground-control points or patches of appropriate distributions.

The third research application of PyLiGram is the 3D transformation of LiDAR blocks based on external matching and adjustment. The results were promising, and the tool worked well. This research is described in a separate paper (which is still under review).

The general conclusion is that PyLiGram is an innovative tool that is based on photogrammetry and ULPI – specific ideas for effective and innovative LiDAR data processing.

6. Future Research

Matching lidargrams and images is the first future research area that was implemented in PyLiGram. The first tests, which were based on external deep-learning matching and adaptation, showed promising results. There are two basic ways to improve the quality of the matching process: the first is to develop a lidargram-generation algorithm and use already-trained deep-learning pipelines; the second is to train a new deep-learning model that is dedicated to LiDAR and image-data matching. A new functionality should be developed within PyLiGram: the automatic generation and analysis of deep-learning samples. The deep-learning-model-building tool for LiDAR and RGB data matching would be trained by using samples that are automatically selected from LiDAR blocks and orthophotos – two data sets that are oriented in the same coordinate system. Future research will include testing the number of samples that are required and the size of the optimal grid step. The next stage of the research and PyLiGram development will include trajectory transformation (or virtual trajectory generation). The process may be based on the photogrammetric resectioning of virtual dense lidargram strips.

It is also planned to develop PyLiGram so that it can be controlled by scripting for more-effective research applications.

Funding

The research is part of a postdoctoral research project supported by the "Excellence Initiative – Research University" program for AGH University of Krakow No. 4744 by Antoni Rzonca.

Author Contribution

A. R. (50%): conceptualization, data curation, formal analysis, funding acquisition, investigation, methodology, project administration, resources, supervision, validation, visualization, writing – original draft, writing – review & editing.

M. T. (50%): data curation, formal analysis, investigation, methodology, resources, software, visualization, writing – original draft, writing – review & editing.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work that is reported in this paper.

Data Availability

Proprietary data, provided for the purposes of the project by private entity and university, available upon prior consent for further use.

Use of Generative AI and AI-Assisted Technologies

No generative AI or AI-assisted technologies were employed in the preparation of this manuscript.

Acknowledgments

We would like to express our gratitude to Prof. Krystian Pyka (AGH University of Krakow) and Prof. Gottfried Mandlburger (TU Vienna) for their valuable suggestions and creative discussions. Furthermore, the real data that was utilized for the testing process was provided by Prof. Gottfried Mandlburger of the Technical University of Vienna, along with Grzegorz Kuśmierz and Dariusz Fryc of Geodimex S.A.

This research constitutes a component of a postdoctoral research project that is receiving partial support from the "Excellence Initiative – Research University" program for AGH University of Krakow (No. 4744). The project is being led by Antoni Rzonca.

References

[1] Baltsavias E.P.: *Airborne laser scanning: Basic relations and formulas*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 54(2–3), 1999, pp. 199–214. https://doi.org/10.1016/S0924-2716(99)00015-5.

[2] Toth C., Jóźków G.: *Remote sensing platforms and sensors: A survey*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 115, 2016, pp. 22–36. https://doi.org/10.1016/j.isprsjprs.2015.10.004.

[3] Cheng L., Chen S., Liu X., Xu H., Wu Y., Li M., Chen Y.: *Registration of laser scanning point clouds: A review*. Sensors (Switzerland), vol. 18(5), 2018, 1641. https://doi.org/10.3390/s18051641.

[4] Li X., Liu C., Wang Z., Xie X., Li D., Xu L.: *Airborne LiDAR: State-of-the-art of system design, technology and application*. Measurement Science and Technology, vol. 32(3), 2020, 032002. https://doi.org/10.1088/1361-6501/abc867.

[5] Bakuła K., Salach A., Zelaya Wziątek D., Ostrowski W., Górski K., Kurczyński Z.: *Evaluation of the accuracy of LiDAR data acquired using a UAS for levee monitoring: Preliminary results*. International Journal of Remote Sensing, vol. 38(8–10), 2017, pp. 2921–2937. https://doi.org/10.1080/01431161.2016.1277044.

[6] Rivera G., Porras R., Florencia R., Sánchez-Solís J.P.: *LiDAR applications in precision agriculture for cultivating crops: A review of recent advances*. Computers and Electronics in Agriculture, vol. 207, 2023, 107737. https://doi.org/10.1016/j.compag.2023.107737.

[7] Dainelli R., Toscano P., Di Gennaro S.F., Matese A.: *Recent advances in unmanned aerial vehicles forest remote sensing – a systematic review. Part II: Research applications*. Forests, vol. 12(4), 2021, 397. https://doi.org/10.3390/f12040397.

[8] Guimarães N., Pádua L., Marques P., Silva N., Peres E., Sousa J.J.: *Forestry remote sensing from unmanned aerial vehicles: A review focusing on the data, processing and potentialities*. Remote Sensing, vol. 12(6), 2020, 1046. https://doi.org/10.3390/rs12061046.

[9] Barclay N., Smith M.: *Assessment of advanced LiDAR based tools for enhanced flood prediction*, [in:] *SoutheastCon 2021*, IEEE, 2021, pp. 1–5. https://doi.org/10.1109/SoutheastCon45413.2021.9401895.

[10] Dreier A., Kuhlmann H., Klingbeil L.: *The potential of UAV-based laser scanning for deformation monitoring – case study on a water dam*, [in:] *5th Joint International Symposium on Deformation Monitoring (JISDM), 20–22 June 2022, Valencia, Spain*, Editorial Universitat Politècnica de València, 2023, pp. 261–269. https://doi.org/10.4995/jisdm2022.2022.13833.

[11] Rashdi R., Martínez-Sánchez J., Arias P., Qiu Z.: *Scanning technologies to building information modelling: A review*. Infrastructures, vol. 7(4), 2022, 49. https://doi.org/10.3390/infrastructures7040049.

[12] Lenda G., Borowiec N., Marmol U.: *Study of the precise determination of pipeline geometries using UAV scanning compared to terrestrial scanning, aerial scanning and UAV photogrammetry*. Sensors, vol. 23(19), 2023, 8257. https://doi.org/10.3390/s23198257.

[13] Azevedo F., Dias A., Almeida J., Oliveira A., Ferreira A., Santos T., Martins A., Silva E.: *LiDAR-based real-time detection and modeling of power lines for unmanned aerial vehicles*. Sensors (Switzerland), vol. 19(8), 2019, 1812. https://doi.org/10.3390/s19081812.

[14] McCoy M.D., Casana J., Hill A.C., Laugier E.J., Mulrooney M.A., Ladefoged T.N.: *Unpiloted aerial vehicle acquired lidar for mapping monumental architecture*. Advances in Archaeological Practice, vol. 9(2), 2021, pp. 160–174. https://doi.org/10.1017/aap.2021.5.

[15] Masini N., Abate N., Gizzi F.T., Vitale V., Minervino Amodio A., Sileo M., Biscione M., Lasaponara R., Bentivenga M., Cavalcante F.: *UAV LiDAR based approach for the detection and interpretation of archaeological micro topography under canopy – the rediscovery of Perticara (Basilicata, Italy)*. Remote Sensing, vol. 14(23), 2022, 6074. https://doi.org/10.3390/rs14236074.

[16] May N.C., Toth C.K.: *Point positioning accuracy of airborne LIDAR systems: A rigorous analysis*. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVI-3/W49B, 2007, pp. 107–111.

[17] Glennie C.: *Rigorous 3D error analysis of kinematic scanning LIDAR systems*. Journal of Applied Geodesy, vol. 1(3), 2008, pp. 147–157. https://doi.org/10.1515/jag.2007.017.

[18] Schaer P., Skaloud J., Landtwing S., Legat K.: *Accuracy estimation for laser point cloud including scanning geometry*. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVI-5/C55, 2007, pp. 1–8.

[19] Bang K.I., Habib A., Kersting A.: *Estimation of biases in lidar system calibration parameters using overlapping strips*. Canadian Journal of Remote Sensing, vol. 36(2), 2010, pp. S335–S354. https://doi.org/10.5589/m10-054.

[20] Li Z., Tan J., Liu H.: *Rigorous boresight self-calibration of mobile and UAV LiDAR scanning systems by strip adjustment*. Remote Sensing, vol. 11(4), 2019, 442. https://doi.org/10.3390/rs11040442.

[21] Lv J., Xu J., Hu K., Liu Y., Zuo X.: *Targetless calibration of LiDAR-IMU system based on continuous-time batch estimation*, [in:] *IEEE International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2020, pp. 9968–9975. https://doi.org/10.1109/IROS45743.2020.9341405.

[22] Lv J., Zuo X., Hu K., Xu J., Huang G., Liu Y.: *Observability-aware intrinsic and extrinsic calibration of LiDAR-IMU systems*. IEEE Transactions on Robotics, vol. 38(6), 2022, pp. 3734–3753. https://doi.org/10.1109/TRO.2022.3174476.

[23] Glira P., Pfeifer N., Mandlburger G.: *Rigorous strip adjustment of UAV-based laserscanning data including time-dependent correction of trajectory errors*. Photogrammetric Engineering & Remote Sensing, vol. 82(12), 2016, pp. 945–954. https://doi.org/10.14358/PERS.82.12.945.

[24] Yuan W., Choi D., Bolkas D.: *GNSS-IMU-assisted colored ICP for UAV-LiDAR point cloud registration of peach trees*. Computers and Electronics in Agriculture, vol. 197, 2022, 106966. https://doi.org/10.1016/j.compag.2022.106966.

[25] Favalli M., Fornaciai A., Pareschi M.T.: *LIDAR strip adjustment: Application to volcanic areas*. Geomorphology, vol. 111(3–4), 2009, pp. 123–135. https://doi.org/10.1016/j.geomorph.2009.04.010.

[26] Gressin A., Mallet C., Demantké J., David N.: *Towards 3D LIDAR point cloud registration improvement using optimal neighborhood knowledge*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 79, 2013, pp. 240–251. https://doi.org/10.1016/j.isprsjprs.2013.02.019.

[27] Tam G.K.L., Cheng Z.Q., Lai Y.K., Langbein F.C., Liu Y., Marshall D., Martin R.R., Sun X.F., Rosin P.L.: *Registration of 3D point clouds and meshes: A survey from rigid to nonrigid*. IEEE Transactions on Visualization and Computer Graphics, vol. 19(7), 2013, pp. 1199–1217. https://doi.org/10.1109/TVCG.2012.310.

[28] Glira P., Pfeifer N., Briese C., Ressl C.: *Rigorous strip adjustment of airborne laserscanning data based on the ICP algorithm*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. II-3/W5, 2015, pp. 73–80. https://doi.org/10.5194/isprsannals-II-3-W5-73-2015.

[29] Brun A., Cucci D.A., Skaloud J.: *Lidar point-to-point correspondences for rigorous registration of kinematic scanning in dynamic networks*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 189, 2022, pp. 185–200. https://doi.org/10.1016/j.isprsjprs.2022.04.027.

[30] Kuçak R.A., Erol S., Erol B.: *The strip adjustment of mobile LiDAR point clouds using iterative closest point (ICP) algorithm*. Arabian Journal of Geosciences, vol. 15(11), 2022, 1017. https://doi.org/10.1007/s12517-022-10303-2.

[31] Csanyi N., Toth C.K., Grejner-Brzezinska D., Ray J.: *Improvement of LiDAR data accuracy using LiDAR specific ground targets*, [in:] *American Society for Photogrammetry and Remote Sensing – Annual Conference 2005 – Geospatial Goes Global: From Your Neighborhood to Whole Planet. Vol. 1*, American Society for Photogrammetry and Remote Sensing (ASPRS), 2005, pp. 152–162.

[32] Chen Z., Li J., Yang B.: *A strip adjustment method of UAV-borne LiDAR point cloud based on DEM features for mountainous area*. Sensors, vol. 21(8), 2021, pp. 1–20. https://doi.org/10.3390/s21082782.

[33] Villena-Martinez V., Oprea S., Saval-Calvo M., Azorin-Lopez J., Fuster-Guillo A., Fisher R.B.: *When deep learning meets data alignment: A review on deep registration networks (DRNs)*. Applied Sciences, vol. 10(21), 2020, 7524. https://doi.org/10.3390/app10217524.

[34] Guo Y., Wang H., Hu Q., Liu H., Liu L., Bennamoun M.: *Deep learning for 3D point clouds: A survey*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43(12), 2021, pp. 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434.

[35] Gupta A., Fernando X.: *Simultaneous localization and mapping (SLAM) and data fusion in unmanned aerial vehicles: Recent advances and challenges*. Drones, vol. 6(4), 2022, 85. https://doi.org/10.3390/drones6040085.

[36] Ressl C., Kager H., Mandlburger G.: *Quality checking of ALS projects using statistics of strip differences*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVII-B3b, 2008, pp. 253–260.

[37] Habib A., Kersting A.P., Bang K.I., Lee D.C.: *Alternative methodologies for the internal quality control of parallel LiDAR strips*. IEEE Transactions on Geoscience and Remote Sensing, vol. 48(1), 2010, pp. 221–236. https://doi.org/10.1109/TGRS.2009.2026424.

[38] Ressl C., Mandlburger G., Pfeifer N.: *Investigating adjustment of airborne laser scanning strips without usage of GNSS/IMU trajectory data*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVIII-3/W8, 2009, pp. 195–200.

[39] Tulldahl H.M., Bissmarck F., Larsson H., Grönwall C., Tolt G.: *Accuracy evaluation of 3D lidar data from small UAV*. SPIE Proceedings, vol. 9649 [Electro-Optical Remote Sensing, Photonic Technologies, and Applications IX], 2015, 964903. https://doi.org/10.1117/12.2194508.

[40] Stöcker C., Nex F., Koeva M., Gerke M.: *Quality assessment of combined IMU/GNSS data for direct georeferencing in the context of UAV-based mapping*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-2/W6, 2017, pp. 355–361. https://doi.org/10.5194/isprs-archives-XLII-2-W6-355-2017.

[41] Dreier A., Janßen J., Kuhlmann H., Klingbeil L.: *Quality analysis of direct georeferencing in aspects of absolute accuracy and precision for a UAV-based laser scanning system*. Remote Sensing, vol. 13(18), 2021, 3564. https://doi.org/10.3390/rs13183564.

[42] Mayr A., Bremer M., Rutzinger M.: *3D point errors and change detection accuracy of unmanned aerial vehicle laser scanning data*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. V-2-2020, 2020, pp. 765–772. https://doi.org/10.5194/isprs-annals-V-2-2020-765-2020.

[43] Fuad N.A., Ismail Z., Majid Z., Darwin N., Ariff M.F.M., Idris K.M., Yusoff A.R.: *Accuracy evaluation of digital terrain model based on different flying altitudes and conditions of terrain using UAV LiDAR technology*. IOP Conference Series: Earth and Environmental Science, vol. 169, 2018, 012100. https://doi.org/10.1088/1755-1315/169/1/012100.

[44] Kucharczyk M., Hugenholtz C.H., Zou X.: *UAV–LiDAR accuracy in vegetated terrain*. Journal of Unmanned Vehicle Systems, vol. 6(4), 2018, pp. 212–234. https://doi.org/10.1139/juvs-2017-0030.

[45] Soininen A.: *TerraMatch User Guide. 64-bit version*. Terrasolid Ltd, Helsinki 2024. https://terrasolid.com/guides/tmatch.pdf [access: 29.06.2025].

[46] RIEGL Laser Measurement Systems GmbH: *RiProcess Datasheet RiProcess Datasheet, 2022-09-01*. https://www.laser-3d.pl/wp-content/uploads/RiProcess_Datasheet_2021-05-17.pdf [access: 29.07.2025].

[47] Lohani B., Ghosh S.: *Airborne LiDAR technology: A review of data collection and processing systems*. Proceedings of the National Academy of Sciences, India Section A – Physical Sciences, vol. 87, 2017, pp. 567–579. https://doi.org/10.1007/s40010-017-0435-9.

[48] López Serrano F.R., Rubio E., García Morote F.A., Andrés Abellán M., Picazo Córdoba M.I., García Saucedo F., Martínez García E., Sánchez García J.M., Serena Innerarity J., Carrasco Lucas L., García González O., García González J.C.: *Artificial intelligence-based software (AID-FOREST) for tree detection: A new framework for fast and accurate forest inventorying using LiDAR point clouds*. International Journal of Applied Earth Observation and Geoinformation, vol. 113, 2022, 103014. https://doi.org/10.1016/j.jag.2022.103014.

[49] Kumar Mishra R.: *A review of optical imagery and airborne LiDAR data registration methods*. Open Remote Sensing Journal, vol. 5, 2012, pp. 54–63. https://doi.org/10.2174/1875413901205010054.

[50] Rzonca A.: *Review of methods of combined orientation of photogrammetric and laser scanning data*. Measurement, Automation & Monitoring, vol. 64(3), 2018, pp. 57–62.

[51] Mouzakidou K., Brun A., Cucci D.A., Skaloud J.: *Airborne sensor fusion: Expected accuracy and behavior of a concurrent adjustment*. ISPRS Open Journal of Photogrammetry and Remote Sensing, vol. 12, 2024, 100057. https://doi.org/10.1016/j.ophoto.2023.100057.

[52] Kovanič Ľ., Topitzer B., Peťovský P., Blišťan P., Gergeľová M.B., Blišťanová M.: *Review of photogrammetric and LiDAR applications of UAV*. Applied Sciences, vol. 13(11), 2023. https://doi.org/10.3390/app13116732.

[53] Chen S., Ma H., Zhang Y., Zhong L., Xu J., Chen H.: *Boresight calibration of airborne LiDAR system without ground control points*. IEEE Geoscience and Remote Sensing Letters, vol. 9(1), 2012, pp. 85–89. https://doi.org/10.1109/LGRS.2011.2161070.

[54] Pentek Q., Kennel P., Allouis T., Fiorio C., Strauss O.: *A flexible targetless LiDAR–GNSS/INS–camera calibration method for UAV platforms*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 166, 2020, pp. 294–307. https://doi.org/10.1016/j.isprsjprs.2020.05.014.

[55] Toschi I., Remondino F., Rothe R., Klimek K.: *Combining airborne oblique camera and LiDAR sensors: Investigation and new perspectives*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-1, 2018, pp. 437–444. https://doi.org/10.5194/isprs-archives-XLII-1-437-2018.

[56] Kalantar B., Ueda N., Al-Najjar H.A.H., Moayedi H., Halin A.A., Mansor S.: *UAV and LiDAR image registration: A SURF-based approach for ground control points selection*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLII-2/W13, 2019, pp. 413–418. https://doi.org/10.5194/isprs-archives-XLII-2-W13-413-2019.

[57] Habib A., Ghanma M., Morgan M., Al-Ruzouq R.: *Photogrammetric and LiDAR data registration using linear features*. Photogrammetric Engineering and Remote Sensing, vol. 71(6), 2005, pp. 699–707. https://doi.org/10.14358/PERS.71.6.699.

[58] Deng F., Hu Minjie, Guan H.: *Automatic registration between LiDAR and digital images*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XXXVII-B1, 2008, pp. 487–490.

[59] Choi K., Hong J., Lee I.: *Precise geometric registration of aerial imagery and LiDAR data*. ETRI Journal, vol. 33(4), 2011, pp. 506–516. https://doi.org/10.4218/etrij.11.1610.0046.

[60] Wang X., Xu B., Zhao S., Li X.: *Automatic registration of optical image and airborne LiDAR data based on centers and corner points of building boundaries*. International Journal of Remote Sensing, vol. 43(18), 2022, pp. 6644–6668. https://doi.org/10.1080/01431161.2022.2142487.

[61] Zhu B., Ye Y., Zhou L., Li Z., Yin G.: *Robust registration of aerial images and LiDAR data using spatial constraints and Gabor structural features*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 181, 2021, pp. 129–147. https://doi.org/10.1016/j.isprsjprs.2021.09.010.

[62] Peng S., Ma H., Zhang L.: *Automatic registration of optical images with airborne LiDAR point cloud in urban scenes based on line-point similarity invariant and extended collinearity equations*. Sensors, vol. 19(5), 2019. https://doi.org/10.3390/s19051086.

[63] Mandlburger G., Wenzel K., Spitzer A., Haala N., Glira P., Pfeifer N.: *Improved topographic models via concurrent airborne LiDAR and dense image matching*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-2/W4, 2017, pp. 259–266. https://doi.org/10.5194/isprs-annals-IV-2-W4-259-2017.

[64] Zahs V., Intertwining L., Anders K., Bremer M., Rutzinger M., Potučková M., Höfle B.: *Evaluation of UAV-borne photogrammetry and laser scanning for 3D topographic change analysis at an active rock glacier*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLIII--B2-2022, 2022, pp. 1109–1116. https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-1109-2022.

[65] Yang B., Chen C.: *Automatic registration of UAV-borne sequent images and LiDAR data*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 101, 2015, pp. 262–274. https://doi.org/10.1016/j.isprsjprs.2014.12.025.

[66] Kim H., Correa C.D., Max N.: *Automatic registration of LiDAR and optical imagery using depth map stereo*, [in:] *2014 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2014, pp. 1–8. https://doi.org/10.1109/ICCPHOT.2014.6831821.

[67] Lv F., Ren K.: *Automatic registration of airborne LiDAR point cloud data and optical imagery depth map based on line and points features*. Infrared Physics & Technology, vol. 71, 2015, pp. 457–463. https://doi.org/10.1016/j.infrared.2015.06.006.

[68] Toschi I., Farella E.M., Welponer M., Remondino F.: *Quality-based registration refinement of airborne LiDAR and photogrammetric point clouds*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 172, 2021, pp. 160–170. https://doi.org/10.1016/j.isprsjprs.2020.12.005.

[69] Li J., Yang B., Chen C., Habib A.: *NRLI-UAV: Non-rigid registration of sequential raw laser scans and images for low-cost UAV LiDAR point cloud quality improvement*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 158, 2019, pp. 123–145. https://doi.org/10.1016/j.isprsjprs.2019.10.009.

[70] Cledat E., Skaloud J.: *Fusion of photo with airborne laser scanning*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. V-1-2020, 2020, pp. 173–180. https://doi.org/10.5194/isprs-annals-V-1-2020-173-2020.

[71] Glira P., Pfeifer N., Mandlburger G.: *Hybrid orientation of airborne LiDAR point clouds and aerial images*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. IV-2/W5, 2019, pp. 567–574. https://doi.org/10.5194/isprs-annals-IV-2-W5-567-2019.

[72] Haala N., Kölle M., Cramer M., Laupheimer D., Mandlburger G., Glira P.: *Hybrid georeferencing, enhancement and classification of ultra-high resolution UAV LiDAR and image point clouds for monitoring applications*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. V-2-2020, 2020, pp. 727–734. https://doi.org/10.5194/isprs-annals-V-2-2020-727-2020.

[73] Haala N., Kölle M., Cramer M., Laupheimer D., Zimmermann F.: *Hybrid georeferencing of images and LiDAR data for UAV-based point cloud collection*

at millimetre accuracy. ISPRS Open Journal of Photogrammetry and Remote Sensing, vol. 4, 2022, 100014. https://doi.org/10.1016/j.ophoto.2022.100014.

[74] Pöppl F., Neuner H., Mandlburger G., Pfeifer N.: *Integrated trajectory estimation for 3D kinematic mapping with GNSS, INS and imaging sensors: A framework and review*. ISPRS Journal of Photogrammetry and Remote Sensing, vol. 196, 2023, pp. 287–305. https://doi.org/10.1016/j.isprsjprs.2022.12.022.

[75] Teo T.A., Shih T.Y., Lin Y.T., Huang C.M.: *The generation of inferred stereo images from LiDAR data,* [in:] *31st Asian Conference on Remote Sensing 2010 (ACRS 2010). Vol. 1*, Asian Association on Remote Sensing (AARS), 2010, pp. 459–464.

[76] Rzonca A., Majek K.: *Lidarometry as a variant of integration of photogrammetric and laser scanning data*. Measurement Automation and Monitoring, vol. 62(8), 2016, pp. 268–273.

[77] Rodríguez-Cielos R., Galán-García J.L., Padilla-Domínguez Y., Rodríguez-Cielos P., Bello-Patricio A.B., López-Medina J.A.: *LiDARgrammetry: A new method for generating synthetic stereoscopic products from digital elevation models*. Applied Sciences, vol. 7(9), 2017, 906. https://doi.org/10.3390/app7090906.

[78] Rzonca A., Twardowski M.: *The lidargrammetric model deformation method for altimetric UAV-ALS data enhancement*. Remote Sensing, vol. 14(24), 2022. https://doi.org/10.3390/rs14246391.

[79] González-Aguilera D., Rodríguez-Gonzálvez P., Hernández-López D., Luis Lerma J.: *A robust and hierarchical approach for the automatic co-registration of intensity and visible images*. Optical Laser Technology, vol. 44(6), 2012, pp. 1915–1923. https://doi.org/10.1016/j.optlastec.2012.01.034.

[80] Jayendra-Lakshman M., Devarajan V.: *A new feature descriptor for LiDAR image matching*. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. II-2/W1, 2013, pp. 157–162. https://doi.org/10.5194/isprsannals-II-2-W1-157-2013.

[81] Fragkos P., Ioannidis C.: *Assessment of lidargrammetry for spatial data extraction*. SPIE Proceedings, vol. 9688 [Fourth International Conference on Remote Sensing and Geoinformation of the Environment], 2016, 96881L. https://doi.org/10.1117/12.2240653.

[82] Tian Z., Wei X.: *Fully convolutional one-stage 3D object detection on LiDAR range images,* [in:] *36th Conference on Neural Information Processing Systems (NeurIPS 2022). Vol. 45*, Advances in Neural Information Processing Systems, vol. 35, Neural Information Processing Systems Foundation, 2022, pp. 34899–34911.

[83] Xianjia Y., Salimpour S., Queralta J.P., Westerlund T.: *Analyzing general-purpose deep-learning detection and segmentation models with images from a lidar as a camera sensor*. arXiv [preprint], 8 March 2022. https://arxiv.org/abs/2203.04064.

[84] Pargieła K., Rzonca A., Twardowski M.: *The utilization of synthetic and semisynthetic point clouds and images for testing novel approaches for correcting LiDAR data*. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XLVIII-1/W3-2023, 2023, pp. 23–24. https://doi.org/10.5194/isprs-archives-XLVIII-1-W3-2023-145-2023.

[85] Schönberger J.L.: *COLMAP*. https://colmap.github.io/ [access: 29.07.2025].

[86] 3DOM-FBK: *Deep Image Matching. Open-source toolbox for multiview image matching*. 3D Optical Metrology (3DOM) Unit, Fondazione Bruno Kessler. https://3dom-fbk.github.io/deep-image-matching/ [access: 29.07.2025].

[87] DeTone D., Malisiewicz T., Rabinovich A.: *SuperPoint: Self-supervised interest point detection and description*, [in:] *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, 2018, pp. 337–349. https://doi.org/10.1109/CVPRW.2018.00060.

[88] Sarlin P.E., DeTone D., Malisiewicz T., Rabinovich A.: *SuperGlue: Learning feature matching with graph neural networks*, [in:] *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2020, pp. 4937–4946. https://doi.org/10.1109/CVPR42600.2020.00499.

[89] Lindenberger P., Sarlin P.-E., Pollefeys M.: *LightGlue: Local feature matching at light speed*, [in:] *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, 2023, pp. 17581–17592. https://doi.org/10.1109/ICCV51070.2023.01616.

[90] Tyszkiewicz M.J., Fua P., Trulls E.: *DISK: Learning local features with policy gradient*, [in:] *34th Conference on Neural Information Processing Systems (NeurIPS 2020). Vol. 18*, Advances in Neural Information Processing Systems, vol. 33, Neural Information Processing Systems Foundation, 2020, pp. 14254–14265.