**Article:**
Bolkas, D., Walton, G., Kromer, R. orcid.org/0000-0002-6036-0919 et al. (1 more author) (2021) Registration of multi-platform point clouds using edge detection for rockfall monitoring. ISPRS Journal of Photogrammetry and Remote Sensing, 175. pp. 366-385. ISSN 0924-2716

https://doi.org/10.1016/j.isprsjprs.2021.03.017

# Registration of multi-platform point clouds using edge detection for rockfall monitoring

**Dimitrios Bolkas[a,*], Gabriel Walton[b], Ryan Kromer[b,c], Timothy Sichler[d]**

[a] Department of Surveying Engineering, The Pennsylvania State University, Wilkes-Barre Campus, Dallas, PA; email: dxb80@psu.edu

[b] Department of Geology and Geological Engineering, Colorado School of Mines, Golden, CO

[c] BGC Engineering, Golden, CO

[d] Department of Electrical Engineering Technology, The Pennsylvania State University, Wilkes-Barre Campus, Dallas, PA

## Abstract

Remote sensing methods which produce point clouds, such as terrestrial laser scanning (TLS), terrestrial photogrammetry (TP), and small unmanned aerial systems (sUAS) have become an integral component of geotechnical monitoring programs. Applications such as rock-slope monitoring benefit from multi-platform datasets to be acquired in two or more different epochs. Accurate registration of these datasets in a common coordinate system is essential for detecting slope changes. Their registration often relies on initial feature-based alignment followed by fine alignment with the iterative closest point (ICP) algorithm. When practical, ground control points (GCPs) and other surveying targets with well-defined coordinates are used. However, establishing such GCPs on rock surfaces can be difficult, expensive and dangerous. In addition, GCPs and targets can be lost or destroyed with time and re-establishing them is difficult. This paper develops an automated registration algorithm based on edge detection that can register multi-platform and multi-epoch point clouds. Rock-surface edges are expected to remain largely the same and be captured in point clouds collected in two different epochs. For edge detection, we use $\alpha$-molecules that offer a unified framework of most multi-scale transforms that can be adapted to any rock-surface. Then the algorithm identifies edge correspondences based on the discrete Fréchet distance. From the corresponding edges we derive matching points between datasets. Transformation parameters are then derived through Procrustes analysis. Using real and simulated scenarios, we demonstrate the utility and performance of the proposed algorithm. For example, sUAS scenarios with 0 and 1 GCPs show that initial root mean square error (RMSE) values of a few decimeters drop to a few centimeters. Scenarios with simulated translations, rotations, and scale showed that the developed algorithm registers multi-platform point clouds with mm differences from their original RMSE values. Results demonstrate that the algorithm can successfully register multi-platform point clouds and support rockfall monitoring.

**Keywords:** point clouds; registration; UAS; laser scanning; monitoring; multi-scale transform

# 1. Introduction

## *1.1 Objectives*

To overcome the limitations of existing registration algorithms for application to complex rock surfaces, this paper proposes the use of a new feature-based algorithm based on natural edges found on rock surfaces. Natural edges are distinct features that are captured in point clouds collected from various platforms and methods, but they have not been utilized in the context of point cloud registration for rockfall monitoring. The objective of this paper is to derive a new feature-based algorithm that can be applied to complex rock surfaces, and is capable of registering multi-platform and multi-epoch point cloud datasets to support rockfall monitoring. The proposed algorithm utilizes corresponding edges present in multi-platform and multi-epoch point clouds. The matched edges are used to identify point correspondences and estimate the transformation between the two datasets.

In addition to their use in the proposed registration algorithm, rock surface edges are important for quantitative analysis and hazard assessment, and their automatic extraction is an active research subject (e.g., Bolkas et al. 2018; Guo et al. 2018; Li et al. 2019). Thus, registration through edge detection offers the potential for creating a unified framework of edge utilization in the context of geotechnical/geological engineering, i.e., from registration to quantitative analysis and hazard assessment. Therefore, utilizing edges for the registration of multi-platform point clouds representing rock surfaces is an interesting research problem worth exploring.

## *1.2 Literature Review*

### *1.2.1 Point clouds in rockfall monitoring*

Terrestrial remote sensing methods have been widely used for rock-slope monitoring. Such methods include terrestrial laser scanning (TLS), terrestrial photogrammetry (TP), and ground-based interferometric synthetic aperture radar (GB-InSAR) (e.g., Abellán et al. 2014; Kromer et al. 2015; Rouyet et al. 2017; O'Banion et al. 2018; Kromer et al. 2019). Despite their significant advantages of high spatial resolution and potential for high temporal resolution in automated monitoring systems, these methods present limitations. Terrestrial data are often collected at a sub-optimal view angle, as setup locations for scanners and/or cameras are limited, thus creating several occlusions, which can lead to significant issues in quantitative analysis. Photogrammetric small unmanned aerial systems (sUASs) have been increasingly used in several mapping and monitoring applications (e.g., O'Banion et al. 2018; Bolkas et al. 2021). Their main advantage is fast and high-resolution data acquisition. In addition, in the case of slope monitoring, sUASs can offer additional information due to the ability to capture images from different vantage locations. Due to these advantages, sUAS-derived point clouds are increasingly used for rock slope monitoring and assessment (e.g., O'Banion et al. 2018; Pérez-Rey et al. 2019).

Monitoring of rock surfaces often uses a single remote sensing method and, when practical, a combination of technologies is used (Letortu et al. 2018; O'Banion et al. 2018; Zieher et al. 2018;

Kromer et al. 2019). Such combinations necessitate the registration of the point clouds in a common coordinate system, both for the cases where datasets are combined in one epoch and between different epochs. This common registration is typically achieved through registration algorithms such as the iterative closest point (ICP) (e.g., Abellán et al. 2010; Van Veen et al. 2017; Williams et al. 2018; Kromer et al. 2019) and using targets with known coordinates or ground control points (GCPs) (Letortu et al. 2018; Zieher et al. 2018; O'Banion et al. 2018). For TLS, such targets do not have to be on the rock surface, as long as there is a line-of-sight connection between the instrument and targets. For photogrammetric methods (TP and sUAS), targets should appear in the captured images, and they are critical to achieving accurate camera self-calibration (O'Banion et al. 2018). Placement of such targets on rock slopes can be dangerous and time consuming. In addition, GCPs are damaged or destroyed with time and re-establishing them is dangerous and difficult. When multi-platform point clouds are not geo-referenced to a common coordinate system (utilizing targets or GCPs), registration relies on data-driven approaches, such as the ICP method (Besl and McKay 1992; Chen and Medioni 1992), which uses common point correspondences extracted from the point clouds. The ICP algorithm has underwent several modifications and improvements through the years. Recent reviews of the successive improvements and variants of the ICP can be found in Dong et al. (2020); Jiang et al. (2021); Zhang et al. (2021).

### *1.2.2 Point cloud registration methods*

Registration of multi-platform datasets is not trivial, as they may have different point-densities, data gaps/overlap, and/or quality. Cheng et al. (2018) provides a comprehensive review of registration methods for laser scanning point clouds. Here, we will briefly describe the main existing methods. Focus is placed on registration methods that act on the final point cloud product and not in any intermediate processing or data acquisition step (e.g., registration of multiple TLS scans, exterior orientation of images). Registration methods are often classified into (i) coarse registration and (ii) refined registration (Rusu et al. 2009). In coarse registration methods, the transformation parameters are unknown, and typically scale is known, as many algorithms assume that the datasets to be registered have the same units. Refined registration methods attempt to fine-tune the registration of two point clouds that are assumed to be closely or approximately registered. In coarse registration methods, corresponding natural features such as points, lines, or planar surfaces are used. For instance, Persad and Armenakis (2017) use wavelet methods to identify keypoints and register airborne and terrestrial point clouds in urban environments. Cheng et al. (2013) and Cheng et al. (2015) use a building corner extraction using a 3D cylinder and point-density for the registration of airborne and terrestrial laser scanning point clouds. Rusu et al. (2009) use a point feature histogram based on the angle between surface normals in TLS point clouds, although, only a visual validation of the algorithm is provided. Other researchers have utilized linear features such as Habib et al. (2005) where linear features are extracted from buildings. They assume that points of the one dataset lie on the line of the second dataset to develop their mathematical model. Their formulation can either directly incorporate lines in the photogrammetric bundle adjustment or use them in a point cloud to point cloud registration. He and Habib (2016) presented a coarse registration method that automatically extracted linear features using a region-growing approach. The extracted linear features are then manually matched to estimate the transformation parameters. In general, coarse registration using linear features depends on the completeness and precision of the extracted linear features (Cheng et al. 2018). Other studies utilized planar features; for example Von Hansen (2006); Brenner et al. (2008) use conjugate planar features in urban environments. However, Brenner et al. (2008) notes that finding suitable planes can be difficult even in urban environments. Often in course registration methods, after coarse alignment of the point clouds, fine alignment is performed using the ICP method. For instance, Novak and Schindler (2013) developed a coarse registration method based on finding the dominant surface normal direction and then using a random sample consensus to derive remaining transformation parameters. Fine registration is then achieved through ICP. Another example is Li et al. (2020b) who created a course registration algorithm based on topological graphs, with an ICP variant for refinement afterwards.

In terms of fine registration methods, Bae and Lichti (2008) developed a variant of the ICP algorithm that uses the surface normal vector and change in curvature to find corresponding points. Other fine registration algorithms have used a point-to-plane approach to register TLS scans (Grant et al. 2012). Cai et al. (2019) developed a method for registering TLS point clouds that have been leveled, thus reducing the number of rotational unknowns to one angle along the z-axis. Given that TLS and photogrammetric methods create points clouds with millions of points, one practical issue that emerges in point cloud registration methods is contamination with incorrect point matches. Accordingly, Cai et al. (2019) reduced the number of outliers of corresponding points in the registration, which were detected using fast point feature histograms (Rusu et al., 2009) and a fast pruning method developed by Bustos and Chin (2017). Another example of registration in the presence of high number of outliers in the corresponding matched points is the graph-enhanced sample consensus, based on graph data structure for point cloud matching (Li et al. 2020a).

### *1.2.3 Challenges in existing registration methods for rockfall detection*

Feature-based registration methods, that rely on natural corresponding geometric features such as points, lines, or planar surfaces have been developed for built environments, and their application in complex rock surfaces is challenging (Abellán et al. 2014). Specifically, in the context of rock surfaces, it is challenging to identify natural corresponding points, linear or planar features (Abellán et al. 2014). In the context of rock surfaces feature-based algorithms based on the scale-invariant feature transform (Lowe 1999), intrinsic shape signatures (Zhong 2009), and point feature histograms (Rusu et al. 2009) have been used to identify ~~points and~~ point correspondences. Such algorithms are used for initial alignment/ registration followed by a fine registration step, usually implemented using the ICP algorithm. For some application examples on rock surfaces see Kromer et al. (2017); Kromer et al. (2019); Xian et al. (2019); Yin et al. (2020). However, such newer feature-based algorithms rely on surrounding characteristics of identified keypoints; however, these might be different in point clouds obtained from different methods (laser versus optical). In addition, Gruen and Akca (2005) developed a least squares 3D surface matching approach, which converges faster than the ICP but needs better initial/approximation values. This algorithm was used in Monserrat and Crosetto (2008) for land deformation. Most registration algorithms are based on TLS-derived point clouds, where scale is usually well defined from the laser measurement system. In point clouds derived from photogrammetric methods, scale is an unknown parameter when GCPs are not utilized. Kromer et al. (2019) developed an automated registration approach to register multi-epoch point clouds generated from a fixed photogrammetric system. The algorithm identified point correspondences using multidimensional feature-based histograms based on surrounding characteristics of each point. Initial scale was estimated using the ratio of the eigenvalues, calculated from points with a correspondence. The registration was then refined through application of the ICP with refined scale adjustment. Despite the many positives of the ICP algorithm (and its variants discussed here), one of its primary limitations is the need for good initial registration of the two input point clouds, as the algorithm might get trapped in local minima and outliers can have a large impact, thus affecting the registration accuracy (Attia and Slama 2017; Li et al. 2020b). This issue is related to the step of finding reliable correspondences between two clouds. In each iteration, point correspondences are identified through a nearest neighbor (closest point) search, which can lead to incorrect point correspondences if the initial alignment is not accurate enough, trapping the algorithm in local minima. Other researchers have pointed out that poor registration results are obtained for low resolution datasets and the algorithm often requires a high number of iterations (Gruen and Akca 2005; Abellán et al. 2014).

### *1.2.4 Edge detection on rock surfaces*

Edges and other features are often used in image matching (Gruen and Stallmann 1993; Gruen 2012). In image analysis, edges are rapid changes in brightness when a grey level contour is crossed (Li 2003), whereas in the spatial domain, edges correspond to rapid changes in the third-dimension of a dataset such as a point cloud representing a rock surface (Bolkas et al. 2018). There are several edge detectors that are often used in image analysis and can be utilized for detecting edges on rock surfaces such as the Sobel, Prewitt, and Canny methods (Prewitt, 1970; Canny 1986; Richards, 2013). Such methods utilize the local gradient (the Sobel and Prewitt methods) or the first derivative of a Gaussian function (the Canny method). In addition to these methods that act in the spatial domain, there are edge detectors that act in the space-frequency domain and perform a multi-scale analysis. Examples of such methods are the wavelet transform, the curvelet transform, the contourlet transform, and the shearlet transform (Shen and Bai 2006; Do and Vetterli 2005; Da Cuhna and Do 2005; Labate et al. 2005; King et al. 2015). Wavelets are not directionally sensitive, which limits their ability to detect edges (Do and Vetterli 2005). Newer transforms such as the Gabor wavelet, curvelets, contourlets, and shearlets were developed to overcome this issue. The above edge detectors were tested in Bolkas et al. (2018) for identifying edges in underground rock surfaces. Results showed that the contourlet and shearlet transforms can be used for automatic edge extraction for rock mass assessment. Recently, Grohs et al. (2016) introduced $\alpha$-molecules, which created a unified framework of most multi-scale transforms (e.g., wavelets, curvelets, shearlets). Reisenhofer and King (2019) extended their shearlet edge detector to $\alpha$-molecules, which is stable in the presence of noise and contrast invariant.

*1.3 Contributions*

The main contribution of this paper is the novel edge-based registration algorithm, which can be used to support rockfall monitoring of multi-platform and multi-epoch datasets. Having a registration method that relies on distinct natural edges facilitates multi-platform registration of point clouds such as point clouds derived from laser scanning and photogrammetric methods. The paper outlines and discusses the main algorithm steps and provides an evaluation of how user-input parameters can affect edge detection, edge matching, and registration.

Additionally, we provide an assessment of the algorithm using multi-platform and multi-epoch datasets (TLS, TP, and photogrammetric-sUAS) from a study site in Idaho Springs, Colorado. We investigate how the proposed registration algorithm performance is connected to the number of GCPs for sUAS photogrammetric surveys. This experiment can highlight the cases where registration algorithms should be employed to enhance registration between datasets.

Finally, we provide application examples of how the proposed registration algorithm can be used to register multi-platform and multi-epoch datasets and support rockfall monitoring for the site in Idaho Springs, Colorado.

The paper is structured as follows: first we present the edge detection and registration algorithm, then we present the study area, the multi-platform and multi-epoch data. Following sections provide the results, numerical assessment of the algorithm, and rockfall detection using the proposed algorithm to register multi-platform and multi-epoch datasets. Finally, conclusions and remarks concerning potential future work are presented.

## 2. Registration Algorithm

The developed registration algorithm relies on identifying corresponding points between two datasets (the target and source datasets). These points are located on identified edges, which should be similar between the two datasets, since they map the same rock formation. Rockfalls that are small in size (e.g., 1-2 $m^3$) can affect edge shape locally, which is why edges used here are longer than 1 m and typically between 2 m and 10 m long. Unless a large rockfall occurs (relative to the size of the study area), this assumption is valid. In the case of a large rockfall, the edges in the rockfall area would become unusable, but such rockfall events would affect any registration algorithm in a similar manner. The algorithm is not suitable for large rockslides or highly degrading slopes, whereby a large proportion of the scene is unstable, and additional investigation is needed to evaluate how the registration algorithm would be affected in such cases. Therefore, the algorithm is based on the logical assumption that similar edges exist in the two multi-platform datasets. The algorithm is tested here for point cloud datasets mapping outcrops and slopes, thus mostly 2.5D point clouds and not fully 3D point clouds.

The first point cloud is the target point cloud that is already georeferenced or used as reference. The second dataset is a point cloud that needs to be transformed to match the target point cloud. In general, the source point cloud needs to be rotated (three rotations), translated (three translations), and scaled (one scale) to match the coordinate system of the target dataset. Registration of TLS point clouds does not need the inclusion of a scale parameter, but photogrammetric point clouds generally do, when GCPs and/or Global Navigation Satellite System (GNSS) are not used. Most examples provided herein do not need a scale parameter, as the TP and sUAS point clouds utilize GCPs (and GNSS for the sUAS images). The algorithm can also accommodate datasets that have different scales, and some examples of application to such cases are provided. In addition, photogrammetric images are affected by shadows, depending on the time of image capture. For instance, the TP data used in this study were captured at 11 am, but the sUAS were captured at 5 pm, because of higher wind and light rain in the early morning and early afternoon hours. The proposed algorithm utilizes the geometric information; thus, it is not directly affected by shadows. However, it is known that photogrammetric point clouds can be noisier in shadow areas than in non-shadow areas.

### *2.1 Algorithm steps*

The main algorithm steps are (Fig. 1): (1) Preparation: rotation through principal component analysis (PCA), phase correlation transformation (PCT), and gridding, (2) edge detection, (3) edge matching, (4) point correspondence identification, (5) Procrustes analysis. These are explained in more detail below. While the algorithm at its current version requires user-input values that are retrieved through a trial-and-error approach, ultimately, generally reasonable parameters (or a range of them) could be identified in the future for use without the need for trial and error. Computational time of the algorithm is about 5 to 6 minutes for the points clouds used here (5.5 to 8.5 million points) and with the following computer hardware: Intel(R) Xeon(R) CPU E5-1650 v4 @ 3.60GHz 3.60 GHz, 64 GB RAM. The most computationally intensive part is the gridding and edge detection steps (3 minutes), edge matching takes 1-2 minutes, and point identification with Procrustes analysis takes 0.5 minutes. The algorithm was developed in Matlab; thus, the algorithm can be optimized and deployed in a more efficient programming language in the future.

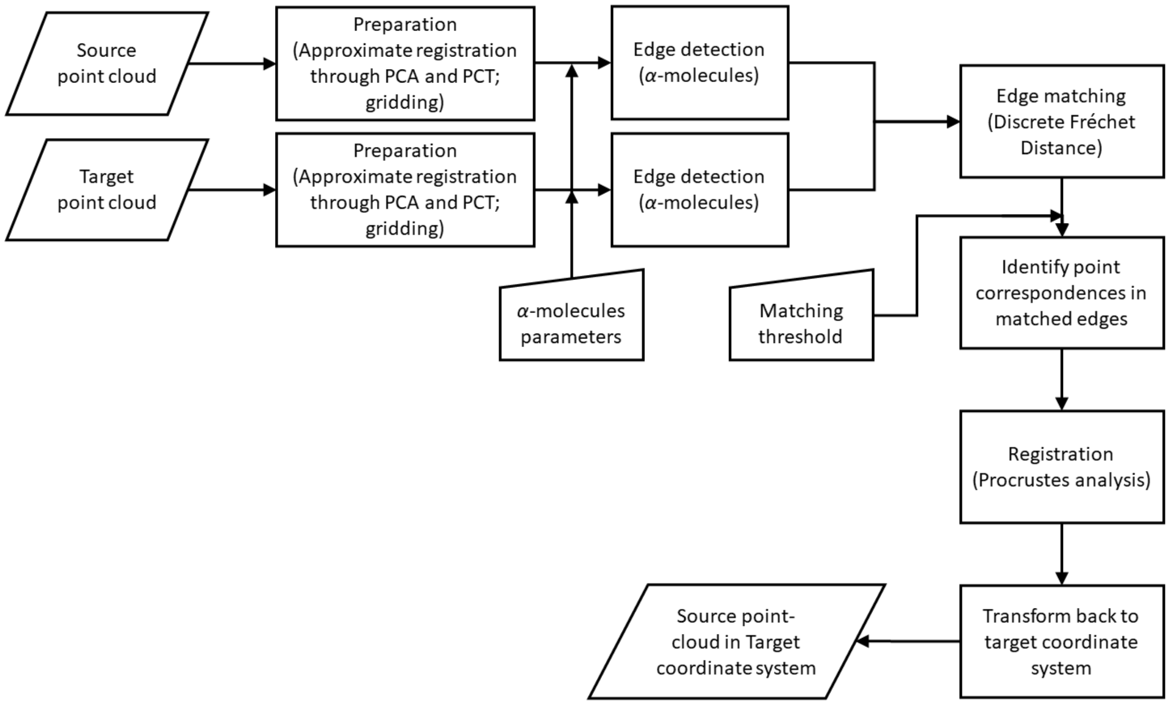



**Fig. 1.** Flowchart showing the main steps of the proposed registration algorithm; boxes with slanted tops indicate input parameters.

<u>(1) Preparation:</u> The point cloud datasets are first rotated using PCA (Jolliffe 2002), making the outcrop span in the x- and y-axes, which facilitates gridding in the edge detection step. Because outcrop datasets are 2.5D, the point clouds will span in the x-y plane after the PCA operation. Small rotations and translations might still exist (e.g., when the two datasets are not cropped to about the same extent); therefore, we apply a PCT to achieve coarse registration (Dimitrievski et al. 2016). To facilitate effective operation of the algorithm, the user should approximately crop the input datasets to the same extent, although rigorous cropping is not crucial (see overlap between datasets for this project in the next section and Figure 5). With approximate cropping of the two datasets, achieving an overlap of at least 70%-80% appears sufficient for acceptable initial registration performance to be achieved. The PCA coefficient matrix, which corresponds to the rotation matrix, is derived using the built-in Matlab function "pca" (MathWorks, Inc. 2020b). The rotation through PCA is given as follows:

$$\mathbf{X}^{S,PCA1} = \mathbf{R}^{S}_{PCA1}\left(\mathbf{X}^{S} - \mathbf{X}^{S,centroid}\right) \tag{1a}$$
$$\mathbf{X}^{T,PCA2} = \mathbf{R}^{T}_{PCA2}\left(\mathbf{X}^{T} - \mathbf{X}^{T,centroid}\right) \tag{1b}$$

where $\mathbf{X}^{S}$ and $\mathbf{X}^{T}$ are the x, y, z coordinates of the source and target point clouds, respectively. $\mathbf{X}^{S,centroid}$ and $\mathbf{X}^{T,centroid}$ are offsets to reduce coordinates to their centroids in order to avoid working with large coordinate values. $\mathbf{R}^{S}_{PCA1}$ and $\mathbf{R}^{T}_{PCA2}$ are 3×3 rotation matrices of the source and target point clouds. The matrices are derived after the PCA operation. $\mathbf{X}^{S,PCA1}$ and $\mathbf{X}^{T,PCA2}$ are the transformed x, y, z coordinates of the source and target point clouds. The PCT (Dimitrievski et al. 2016) is given as:

$$\mathbf{X}^{S,PCT} = \mathbf{R}^{S}_{PCT}(\mathbf{X}^{S,PCA1} + \mathbf{X}^{S,PCT}) = \begin{bmatrix} \cos\theta^{PCT} & -\sin\theta^{PCT} & 0 \\ \sin\theta^{PCT} & \cos\theta^{PCT} & 0 \\ 0 & 0 & 1 \end{bmatrix} \left( \mathbf{X}^{S,PCA1} + \begin{bmatrix} T_x^{PCT} \\ T_y^{PCT} \\ 0 \end{bmatrix} \right) \tag{2}$$

Where $\mathbf{X}^{S,PCT}$ is the source point cloud transformed using the PCT method, $\mathbf{R}^{S}_{PCT}$ is the rotation matrix around the z-axis with angle $\theta^{PCT}$, and $\mathbf{X}^{S,PCT}$ the translation vector for the x- and y-axis with $T_x^{PCT}$ and $T_y^{PCT}$ translations, respectively. The sub- and super-script $PCT$ means that the rotation and translation values have been derived from the phase correlation transformation. Note that if the two datasets are already approximately georeferenced (e.g., within few meters) and have similar extent, such as some of the datasets in this paper, then Eq. (1) can be simplified by using the same centroid values and rotation matrix as follows: $\mathbf{X}^{S,centroid} = \mathbf{X}^{T,centroid}$ and $\mathbf{R}^{S}_{PCA1} = \mathbf{R}^{T}_{PCA2}$, and the PCT step in Eq. (2) can be discarded.

The rotation to the x-y plane facilitates gridding of the point clouds in order to utilize edge detection algorithms. Gridded point clouds can be treated like a grayscale image, allowing us to utilize edge detection methods that were originally designed for images. The grid-step is an input parameter and depends on the point cloud resolution. A higher resolution dataset enables better identification of edge location, although depending on the size of the study area, could lead to large grids that increase computational burden. In this study, grid spacings of 10 cm, 5 cm, and 2.5 cm were tested for comparison. A Delaunay triangulation and linear interpolation is used to grid the point clouds. The algorithm can produce grids with different spatial extent and grid spacing, although if the two datasets have the same scale, then the same grid spacing is used for consistency. From the transformed point clouds of Eq. (1) and Eq. (2) we derive three arrays for each dataset (i.e., one array for each coordinate component). The z-values of the two gridded datasets are then converted to gray-scale images. Note that in the presence of significant scale differences between the two datasets, different grid steps are used for the target and source datasets, based on their average point densities. Scale and grid step are then refined trough iteration. Estimation of initial scale is further discussed in: (3) edge matching.

<u>(2) Edge detection:</u> In the next step, edge detection is performed using a symmetric $\alpha$-molecules approach (Reisenhofer and King 2019). The main goal of this step is to identify the same edges in the two datasets (Fig. 2). Symmetric $\alpha$-molecules were selected because they offer a general formulation that can be adapted to resemble wavelets, curvelets, and shearlets (Reisenhofer and King 2019). In addition, several edge detection algorithms were tested in Bolkas et al. (2018) for edge detection on rock formations, which showed the superiority of shearlets over several edge detection methods. In this step, the user needs to specify input parameters that can be derived empirically (see for example Bolkas et al. 2018). These input parameters rely on the rock formation itself and not so much on the point cloud creation method (i.e., laser scanning or imagery). Based on previous experience from Bolkas et al. (2018), we have derived ranges of parameter values that can work in this type of application case, and they are discussed below.

The edge detection based on $\alpha$-molecules is implemented using the SymFD toolbox (Reisenhofer and King 2019). The main user input parameters are organized in the following eight main categories: (1) minimum and maximum feature width, which determine the width of the narrowest and widest filters, respectively, in the system of symmetric molecules; (2) The maximum feature length, which determines the length of the longest filter in the system. The maximum length and maximum width specify the length and width of the generating filter; (3) The generator, which in this study is the tensor product of a Mexican hat wavelet and a Gaussian Filter (as used in Bolkas et al. 2018); (4) The $\alpha$ parameter, which determines the degree of anisotropy of the applied scaling matrix, where $\alpha$=1 is isotropic scaling and $\alpha$=0 is anisotropic scaling; (5) the number of scales; (6) The number of differently oriented molecules on each scale; (7) An option for if a rotation matrix will be used or if a shear matrix will be used (see King et al. 2015; Reisenhofer et al. 2016); (8) The minimal contrast for edge detection.

Because there are a quite large number of possible input values for the above parameters, we have limited testing to few values utilizing the knowledge gained through trial and error and from Bolkas et al. (2018); these are: (1) minimum feature width of 1 m and 2 m, maximum feature width of 4 m, 6 m, and 8 m; (2) maximum feature length 4 m, 6 m, and 8 m; (3) as described above, the generator is the tensor product of a Mexican hat wavelet and a Gaussian Filter as used in Bolkas et al. (2018); (4) $\alpha$ values of 0, 0.5, and 1; (5) 2, 4, and 6 scales; (6) 2^6, 2^8, 2^12 and 2^16 numbers of orientations; (7) both the rotation and shear matrix conditions; and (8) minimal contrast values of 1 and 2. Note that in preliminary testing, we considered a 2 m maximum feature length and width, but this value led to fragmented edges and high computation load compared to the 4 m (and higher value) scenarios, so 4 m is the smallest feature width and length considered here.

In Bolkas et al. (2018) an empirical approach was implemented to search and find the parameters that provide the best detector by comparing with manually extracted edges (used as reference). Here, edges do not have to match with a reference edge dataset, but rather we want the detector to detect edges in the same location for two input multi-platform datasets. To assess the detector and find the edge parameters that will give us the most similar edges, we are assessing the agreement (in detected edges) between the three input datasets (i.e., between the TP and sUAS datasets, between the TLS and sUAS datasets, and between the TP and TLS datasets). Users are expected to conduct a visual comparison because it is assumed that their datasets will be in different coordinate systems and thus quantitative assessment cannot be performed. For the purposes of this study, a comparison is possible because all three datasets are geo-referenced with cm-level agreement. The relative edge agreement is verified using a simple metric, Pratt's Figure Of Merit (PFOM) (Pratt 1978), which is defined as:

$$\mathrm{PFOM} = \frac{1}{max\{E_A, E_D\}} \sum_{i=1}^{E_D} \frac{1}{1 + f d(i)^2} \tag{3}$$

where $E_A$ is the number of actual edge points, $E_D$ is the number of detected edge points, $d(i)$ is distance (in pixels) between the actual edge to the corresponding detected edge of the $i^{\text{th}}$ detected edge point, and $f$ is a scaling constant chosen as 1/9. The metric takes on values from 0 to 1, with 1 corresponding to perfect edge similarity between the two compared edge datasets.

The output of the edge detection operation for each dataset is a binary array with values of zeros and ones, where a value of one indicates the presence of an edge grid cell and a value of zero indicates its absence (i.e., $\mathbf{E}^S$ and $\mathbf{E}^T$ for the source and target datasets, respectively). From the binary edge images, we extract lists of connected edge points using the algorithm of Kovasi (2019).

The lists consist of sets of integer (row and column) locations of the connected edge points. From the row and columns lists we can extract the coordinates of the identified edges in the source and target datasets:

$$\mathbf{L}^S = \begin{bmatrix} \mathbf{L}_1^S \\ \vdots \\ \mathbf{L}_p^S \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^{S,grd} & \mathbf{y}_1^{S,grd} & \mathbf{z}_1^{S,grd} \\ \vdots & \vdots & \vdots \\ \mathbf{x}_p^{S,grd} & \mathbf{y}_p^{S,grd} & \mathbf{z}_p^{S,grd} \end{bmatrix} \tag{4a}$$

$$\mathbf{L}^T = \begin{bmatrix} \mathbf{L}_1^T \\ \vdots \\ \mathbf{L}_q^T \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1^{T,grd} & \mathbf{y}_1^{T,grd} & \mathbf{z}_1^{T,grd} \\ \vdots & \vdots & \vdots \\ \mathbf{x}_q^{T,grd} & \mathbf{y}_q^{T,grd} & \mathbf{z}_q^{T,grd} \end{bmatrix} \tag{4b}$$

where $\mathbf{x}_p^{S,grd}$, $\mathbf{y}_p^{S,grd}$, $\mathbf{z}_p^{S,grd}$ contain the real-world coordinates of the grid-cells that comprise the $p$th edge in the source dataset and $\mathbf{x}_q^{T,grid}$, $\mathbf{y}_q^{T,grid}$, $\mathbf{z}_q^{T,grid}$ contain the real-world coordinates of the pixels that comprise the $q$th edge in the target dataset. Note that in the general case, the number of identified edges will be different in the source and target datasets (i.e., $p \neq q$). The $grid$ superscript denotes gridded coordinates.

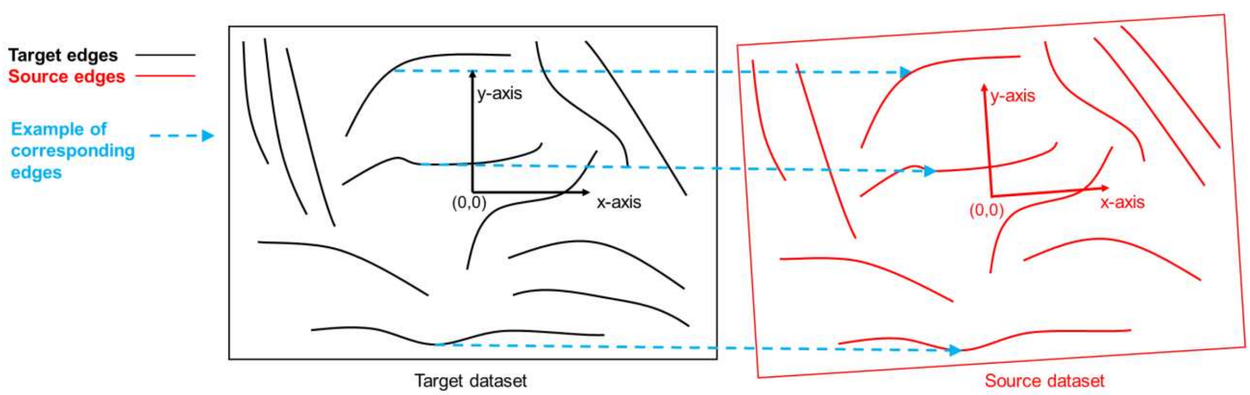



**Fig. 2.** Outline of target and source datasets after PCA and PCT. The figure shows detected edges for the target and source datasets. Also, the figure shows an example of corresponding edges in the two datasets.

(3) Edge matching: The detected edges from the source dataset are matched with the detected edges of the target dataset. The process is based on the Discrete Fréchet Distance (DFD) (Eiter and Mannila 1994) as implemented by Danziger (2021). The DFD is a measure of similarity between two polygonal curves, and it is often used to measure the similarity between two trajectories (e.g., Zhang et al. 2019). The DFD is often explained with the example of a man walking a dog on a leash (Eiter and Mannila 1994). The man is moving on the one curve and the dog on the other. Their traversing speed can vary, but no backtracking is allowed. The DFD measures the shortest "leash" length for traversing the two curves. The algorithm considers the location and ordering of points in the two curves, and it can compare curves of different number of points. We compute the DFD between all possible pairs of the detected edges in the source and target datasets. For example,

considering two edges $\mathbf{L}_p^S$ and $\mathbf{L}_q^T$, one in the source dataset and in the target dataset, the DFD is given as (Eiter and Mannila 1994):

$$\mathrm{DFD}(\mathbf{L}_p^S, \mathbf{L}_q^T) = \min_{C \in \Omega} \max_{(\mathbf{L}_p^S(a_i), \mathbf{L}_q^T(b_i)) \in C} \left\| \mathbf{L}_p^S(a_i) - \mathbf{L}_q^T(b_i) \right\| \qquad (5)$$

Where $\Omega$ is the set of all possible couplings between $\mathbf{L}_p^S$ and $\mathbf{L}_q^T$, $C$ is a coupling between $\mathbf{L}_p^S$ and $\mathbf{L}_q^T$, which is a sequence $\left(\mathbf{L}_p^S(a_1), \mathbf{L}_q^T(b_1)\right), \left(\mathbf{L}_p^S(a_2), \mathbf{L}_q^T(b_2)\right), \ldots, \left(\mathbf{L}_p^S(a_m), \mathbf{L}_q^T(b_m)\right)$ of distinct pairs between the points of the two curves. The sequence respects the ordering of the points in the two curves such that $a_1 = 1$, $b_1 = 1$, $a_m = p$, $b_m = q$, and for all $i = 1, \ldots, q$ we have $a_{i+1} = a_i$ or $a_{i+1} = a_i + 1$, and $b_{i+1} = b_i$ or $b_{i+1} = b_i + 1$. Figure 3 shows an illustration of the DFD in cases where the edges are similar (a vertical offset was applied for illustration in Fig. 3a) and in cases where the edges are not similar, demonstrating the longer "leash" or coupling distance that is needed when the edges are not corresponding.

The application of DFD in the detected edges of the source and target datasets results in a $p \times q$ matrix of DFD distances. For each target edge, we find the source edge with the smallest DFD. For duplicate matches we keep the ones with the smaller DFD. Then we keep the strongest edge correspondences between the two datasets by applying a threshold. The DFD considers both the location and ordering of points on two curves /edges; thus, the metric is affected by the location of edges and an approximate initial registration is required for edge matching. The accuracy of the initial registration depends on the spacing between edges. In this case, edges are several meters apart; therefore, the initial registration should align the two datasets at the 1-2 meter level or better, otherwise the edge matching step can produce poor matches. In that case, the user would need to refine the PCT parameters or manually align the two point clouds.

Furthermore, the DFD finds a one-to-one match with the minimum DFD value. In the case where an edge is long in the first dataset and the corresponding edge is fragmented into smaller edges in the second dataset, only one of the fragmented segments (the one with the minimum DFD value) will be matched to the long edge. To avoid obtaining fragmented edges, the user can select appropriate max feature width and max feature length values in the edge detection step with $\alpha$-molecules. For instance, in this example we do not use max feature width and length parameters of less than 4 m, such parameters led to fragmented edges.

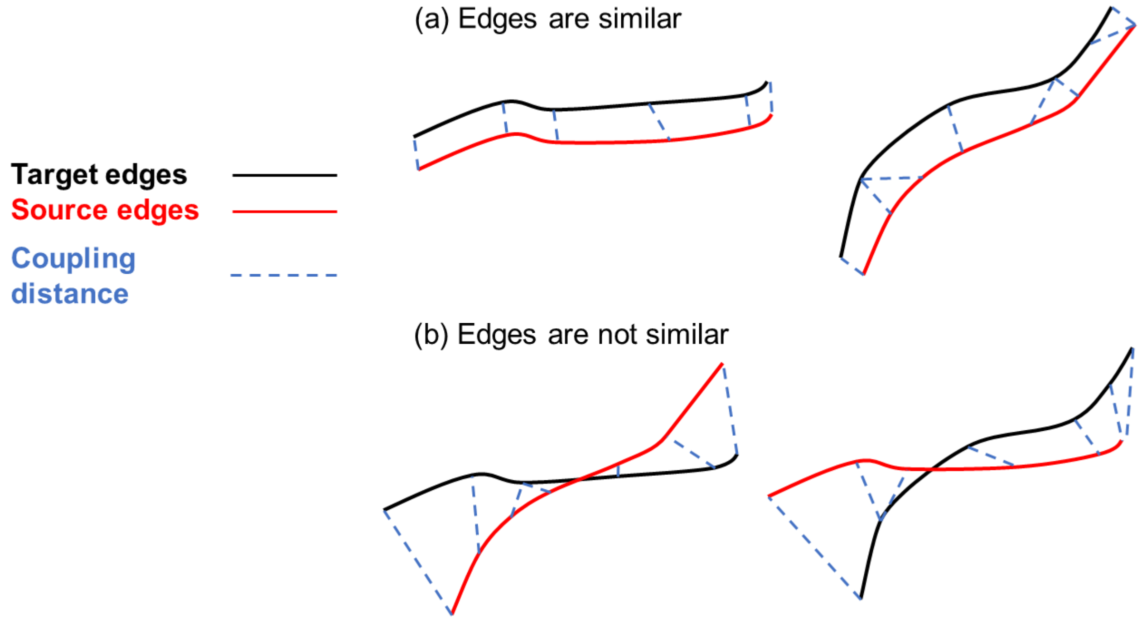



**Fig. 3.** Illustration of the Discrete Fréchet Distance in cases where (a) edges are corresponding and (b) edges not corresponding between the target and source datasets.

Computational time increases with increasing number of edges. About 500 to 600 edges larger than 1 m are identified for each dataset. Utilizing parallel computing, the DFD estimation for all possible pairs took about 2.5 minutes in our computer system. We have also developed a variant where we compute the DFD only for source edges that are within a radius of the midpoint of the reference edges (e.g., 5 m or 10 m), which speeds the process to less than 1 minute of processing time without parallel computing. Results are identical in the two cases, with the only difference being the processing time.

*Note on Scale*: In the case of significant scale between the two datasets (e.g., in the case of photogrammetric point clouds with no GNSS or GCPs), the DFD will not be accurate. A way of retrieving initial estimates of scale is by computing the singular value decomposition (SVD) of the two datasets and using the ratio of the singular values between the two datasets (Umeyama 1991). Another approach is to utilize the 2.5D nature of the datasets and use the ratio of the standard deviations in the z-direction (after the PCA rotation) for the two datasets. We have found that both these approaches yield similar estimations of initial scale value. With an initial estimation of scale, the algorithm computes average point densities to derive a grid step for the source dataset. A new scale is then estimated in the Procrustes analysis step. Scale and grid step, as well as the input parameters for edge detection (maximum feature length and width, and minimum feature width), are then refined for the source dataset through iteration.

(4) Identify point correspondence: At this point, we have identified matching edges in the two datasets. Next, we derive corresponding points in the identified edges (Fig. 4). Point correspondences are achieved by computing the sum of squared differences (SSD) between points of the two matched edges and using that as a criterion to identify correspondences. We use the 3D gridded coordinates for the point-matching step (hence the term 3D gridded points in Fig. 4). The matching is implemented using the Matlab function "matchFeatures" (MathWorks, Inc. 2020a), which uses the sum of squared differences as metric. A max ratio threshold, with values between zero and one, is selected by the user to identify the strongest matches and reject ambiguous matches. After this step, the algorithm should have identified matching points in the two datasets that can be used to estimate a transformation. Unmatched points ("weak" correspondences) exist because of higher differences in coordinates than matched points ("strong" correspondences), which can be due to actual point cloud differences between the target and source datasets or due to differences in detected edge-points from the edge detection step. Examples of such "weak" correspondences are depicted in the source dataset in Fig. 4.

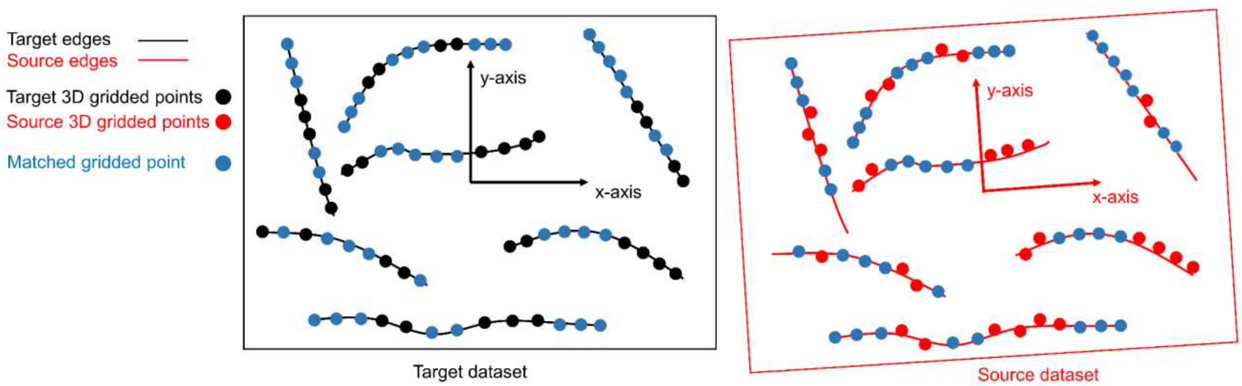



**Fig. 4.** Point correspondences in the target and source datasets. The circles show locations of 3D (gridded) points that make the identified edges.

(5) Procrustes analysis: The transformation estimation utilizes 3D coordinates, with 3D coordinates taken from their gridded values. We use a Procrustes analysis that determines the linear transformation (scale, three rotations and three translations) of the points in the one dataset that best conforms to the shape of the second dataset (Kendall 1989; Dryden and Mardia 1998), thus maintaining the relative shape formed by the source-matched points. The residuals are assessed to remove outliers and incorrectly matched points. A threshold of two standard deviations has been used in most cases here after trial and error. Although, this approach was sufficient here, another approach that could be considered in the future is to use robust outlier removal methods in the case of highly contaminated matches, such as the guaranteed outlier removal method (Bustos and Chin 2017) and the graph-enhanced sample consensus (Li et al. 2020a). The transformation that transforms the source dataset to the coordinate system of the target dataset is then given as:

$$\mathbf{X}_{PA}^{S,PCAT} = \mathrm{b}\mathbf{R}_{S,PCT}^{T,PCA2}\mathbf{X}^{S,PCT} + \mathbf{T}_{S,PCT}^{T,PCA2} \tag{6}$$

where b is the scale difference between the source and target datasets, $\mathbf{T}_{S,PCT}^{T,PCA2}$ is the translation component from the PCT space of the source dataset to the PCA space of the target dataset, $\mathbf{R}_{S,PCT}^{T,PCA2}$

is the rotation component from the PCT space of the source dataset to the PCA space of the target dataset, and $\mathbf{X}_{PA}^{S,PCA2}$ are the transformed source coordinates to the PCA space of the target dataset. The transformation takes place in the PCA space; therefore, to derive the object space coordinates, $\mathbf{X}_{PA}^{S,PCA2}$ needs to be rotated by $(\mathbf{R}_{PCA2}^{T})^{-1}$ and translated by $\mathbf{X}^{T,centroid}$.

*2.2 Algorithm assessment*

Assessment of the proposed algorithm starts with an investigation of the "best" input parameters for edge detection with $\alpha$-molecules, and we illustrate the ability of the edge detector to identify corresponding edges in multi-platform point clouds. We evaluate the edge matching step and investigate the number of correct / incorrect matched edges based on the DFD threshold. In addition, we assess the effect of the number of identified corresponding points on the registration result (Procrustes step). Registration / georeferencing of photogrammetric surveys relies on the number of GCPs considered, so we test several scenarios with varying number of GCPs to highlight the cases where the proposed algorithm can contribute in the registration of photogrammetrically derived point clouds.

Comparisons with the ICP algorithm as implemented in Cloud Compare (CloudCompare 2015) are provided. An ICP implementation using correspondence rejection based on the Point Cloud Library (Holz et al. 2015) was used to validate the ICP results. The root mean square error (RMSE) values were found to be within 1-2 mm of the Cloud Compare implementation. Therefore, for this study area, the RMSE values obtained from the Cloud Compare ICP are considered to provide a good reference comparison. Furthermore, we test the developed algorithm with simulated registration scenarios of multi-platform point clouds with significant translations, rotations, and scale. The last assessment section uses multi-epoch scenarios to show how the proposed algorithm can be used to register multi-epoch and multi-platform point clouds to support rockfall monitoring.

To numerically assess the registration algorithm, we compute RMSE values between the multi-platform datasets, using the model-to-model cloud comparison (M3C2) algorithm (Lague et al. 2013; CloudCompare 2015). In this study, a point spacing of 10 cm was used, meaning that core points that are for calculation of the M3C2 distance between two input point clouds had an average point spacing of 10 cm (Lague et al. 2013; CloudCompare 2015).

Most multi-platform comparisons consider the TP, TLS and sUAS datasets from 2019 data acquisition, which eliminates the effect of active rockfalls in the RMSE values. Multi-temporal registration scenarios are also considered to assess whether the algorithm can support point cloud registration for change detection. To evaluate the algorithm's performance, RMSE values from ICP-based registrations are used as a benchmark. From the M3C2 results, we compute a RMSE value to assess the agreement or disagreement between two multi-temporal or two multi-platform point clouds. Points in vegetated areas are removed before computing the RMSE value using a manually made vegetation mask to eliminate their effects, as they correspond to lower accuracy areas in point cloud data (e.g., Gruszczyński et al. 2017; Bolkas 2019).

## 3. Study area and data

The study area is located west of Idaho Springs, Colorado along interstate I-70. The site is monitored using terrestrial photogrammetry data from the Colorado Department of Transportation and The Colorado School of Mines (see Kromer et al. 2019). The area under investigation is 55 m wide and 35 m in height (Fig. 5a). This area is part of the front range of the Colorado Rocky Mountains and it consists of jointed biotite gneiss. The biotite gneiss is characterized by interlayers of plagioclase-biotite gneiss and sillimanitic biotite-quartz gneiss and is well foliated (for more information, see Kromer et al. 2019). The region and site under investigation is prone to rockfalls due to the steep canyons and seasonal freeze and thaw cycle (Kromer et al. 2019).

This study uses multi-platform point clouds acquired from terrestrial photogrammetry, aerial photogrammetry (from sUAS), and terrestrial laser scanning. A GNSS network was created to provide geo-referencing of the various datasets. These are presented in detail below. The TP images used here were acquired in March of 2018 and June of 2019; the TLS point clouds were collected in February of 2018 and June of 2019; the sUAS data were collected in June 2019 (see Figure 5h). For the sUAS data there is not a 2018 acquisition, because the system was not available at that time. Figures 5e-5g show the various data on top of the sUAS dataset, to highlight their spatial extent. Note that the datasets were moved for illustration.

### *3.1 GNSS network*

To register and georeference the TLS scans, a network of 11 points was established with GNSS observations (triangles in Fig. 5a). These points were one-meter rebars fit into the ground. The baselines of the 11 points were observed with 10-minute rapid static observations using two Leica GS 15 GNSS units. The coordinates were referenced with respect to NAD 83 (2011) State Plane Coordinates, Colorado Central Zone using a National Geodetic Survey Continuously Operating Reference Station (CORS) that was 30 km away from the site location. The referencing was accomplished by solving the baseline from the said point to one of the GNSS control points (triangle with black outline in Fig. 5a) using a 2-hour static survey. Only one point was used as a reference for georeferencing to avoid distortion of the GNSS network (minimally constrained network).

Seven points are located on the opposite side of the site location and interstate I-70, as scanning very close to the rock formation does not offer an optimal vantage scanning location. Four points were located next to the rock formation to assist with registration if needed (Fig. 5a), but they were not ultimately used. The average post-adjustment standard deviation was 5 mm for Easting and Northing coordinates and 1 cm for ellipsoidal height. The ellipsoidal height standard deviation did not exceed 1 cm in the seven points located on the opposite side of the rock formation. Instead, the four points located next to the rock formation had the highest standard deviations. These were up to 1 cm for Easting and Northing and 1.3 cm to 3.0 cm for ellipsoidal height. The higher standard deviations are justified as the points are very close to the rock formation, limiting satellite view. This was another reason why these points were not used in the TLS registration, to ensure high quality registration.

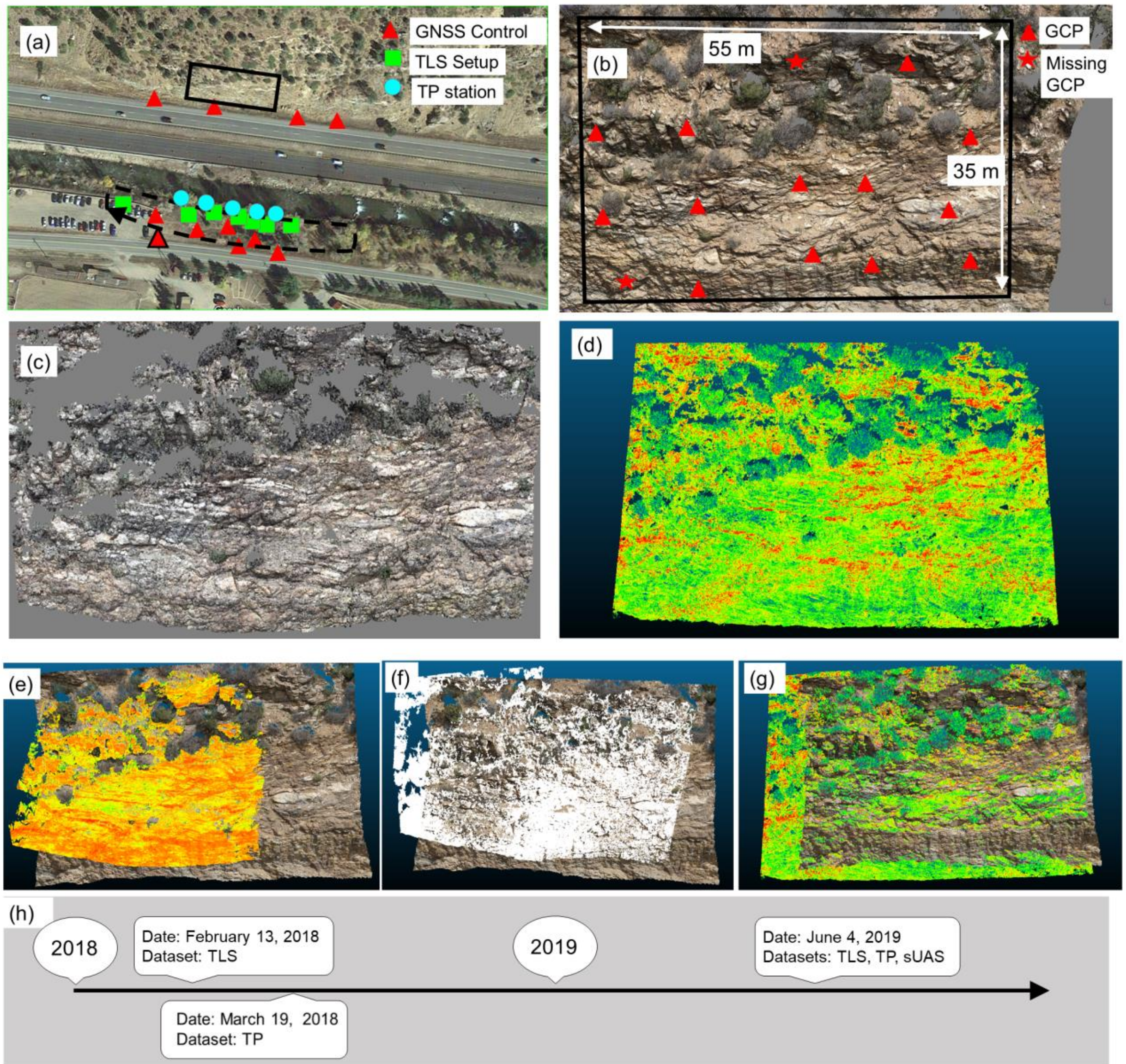



**Fig. 5.** Study area, data acquisition, and datasets. (a) GNSS control points, TLS setup locations, TP stations, study area (solid line) and sUAS trajectory (dashed line). The triangle with black outline shows the point that was used to georeferenced the network with respect to NAD 83 (2011) (b) GCPs on the rock-surface for the photogrammetric surveys, triangles show found GCPs, and stars show destroyed/lost GCPs. The point cloud from the sUAS survey of 4 June 2019 is also shown (c) TP dataset from 4 June 2019 (d) TLS dataset from 4-5 June 2019; (e) TLS from 2018 on top of sUAS from 2019; (f) TP from 2019 in white on top of sUAS from 2019; (g) TLS from 2019 with sUAS from 2019; (e) graphical timeline. Note that for figures (e), (f), and (g) the datasets were moved for illustration of overlap between the different datasets.

*3.2 TLS data*

The squares in Fig. 5a show the scanning setups and Fig. 4d shows the resulting TLS point cloud. Finding setup locations was challenging due to the high vegetation and tree branches. Scanning from the road median was not an option due to safety considerations. Ultimately, laser scanning data were collected from 7 locations on the opposite side of the highway. The distance between the scanning locations and rock formation is between 80 m to 100 m (lower to higher part of the rock formation).

In the data acquisition of June 4-5 2019, a Leica Scanstation P40 was used. The scanner has a range precision of 1.2 mm + 10 ppm, angular precision of 8´´, and a 6 mm 3D position accuracy at 100 m (for the instrument's noise performance in various targets see Bolkas and Martinez 2018). The field scanner resolution was set to 0.8 mm at 10 m and the average point spacing after the data from all scan locations were merged was approximately 1 cm. The average point spacing was unified to 2.5 cm after registration to more closely match the point cloud footprint size on the outcrop (about 2-2.3 cm based on a 0.23 mrad beam divergence) and to match the average ground spacing of the sUAS dataset (2.2 cm). The seven points in Fig. 5a were used for point cloud registration using resection. Each scanner setup utilized at least three points for the resection, except for the rightmost square on Fig. 5a, which used only two. Coordinate standard deviations of the resection did not exceed 1 cm for all scanner setups.

Another TLS dataset from February 13 of 2018 was used as a reference for rock-fall detection. The dataset was collected with a Faro Focus x330 phase shift scanner. The scanner has a ranging error of 2 mm, and a beam divergence of 0.19 mrad. The scanner was set on the median of the highway, which is closer than the June 2019 data acquisition, hence the resulting point cloud had a 1 cm average point spacing.

*3.3 sUAS data and GCPs*

The Colorado Department of Transportation (CDOT) placed 15 ground control points on the rock surface in 2018 (Fig. 5b) (Kromer et al. 2019). Of these 15 points, 13 points were found in May of 2019 (triangles in Fig. 5b; stars show the two destroyed/lost points). Their coordinates were estimated in the same coordinate system as the GNSS network discussed above through total station observations. A Leica TS 15 was used that has a 1´´ angular precision and 2 mm + 2 ppm range precision for any surface. The total station was setup on the GNSS control points. Observations were collected from at least two different locations for each GCP of the rock surface. Observations were collected in two sets (direct and reverse instrument mode) from each setup. Standard deviations of coordinate differences ranged from 2 mm to 11 mm, thus confirming high precision.

Aerial imagery was collected using an Aibotix X6 unmanned aerial vehicle and using a Sony Alpha a6000 mirrorless digital camera. The camera has a 24.3-megapixel sensor and a 16-50 mm lens. The sUAS has a GNSS capability that reduces the required number of control points. In Bolkas (2019) and Bolkas et al. (2019) it was found that 4-6 GCPs are sufficient for cm-level accuracy when the sUAS utilizes its GNSS-Real Time Kinematic (RTK) capability, and about 12 GCPs are required when georeferencing relies on GCPs only. A GNSS base station was set on one of the control points; therefore, the 13 GCPs are more than the minimum required to provide cm-level georeferencing.

Flights were conducted at an altitude of 30 m using focal lengths of 16 mm, 25 mm, and 30 mm. Each focal length has two flights: one flight on June 4, 2019 and one on June 5, 2019. The same trajectory was followed for all flights and is shown as a dashed line in Fig. 4a. Oblique imagery was captured instead of nadir to maintain a vantage line-of-sight with the rock surface. The arrow is at the approximate take-off location and shows the direction of flight. Flights closer to the rock surface were not possible due to restrictions associated with the interstate.

For the purposes of this study, the 30 mm focal length-flight of June 4, 2019 is used, which had 35 images. The 30 mm flight is used because it offers images with better resolution than the 16 mm and 25 mm flights. The corresponding size of image pixels on the ground for this flight is 1.1 cm. The aerial images were processed in Agisoft Metashape (Agisoft LLC 2020) using the “high” quality level, which generated a point cloud with 2.2 cm average point-spacing. The quality level (ultra, high, medium, low, lowest) specifies the dense point cloud reconstruction quality in Agisoft Metashape. The ultra-high quality level processes the original photos, while each other level downscales input images by a factor of four (Agisoft LLC 2020). This reduces processing times, but results in decreased point cloud density. The 2.2 cm average point spacing obtained from the high quality level is considered sufficient for this study.

### *3.4 TP data*

Terrestrial photogrammetry data were acquired using the fixed station camera system established on the site and developed by Kromer et al. (2019). There are five camera stations installed at the opposite side of the slope. The cameras used are Canon 5DSRs with 50-megapixel resolution. Each camera station is 10-15 m apart and they are approximately parallel to the slope (see Fig. 4a and Kromer et al. 2019). The system automatically triggers the camera to capture photos two times a day (at noon and at 12:30 pm for redundancy), processes the TP point clouds using Agisoft Metashape via python scripting, and registers and calculates change using custom-built algorithms. The system works without GCPs as it possesses accurate pre-calibration models for the digital cameras. The GCPs were only used as a comparison for the GCP-less method in Kromer et al. (2019).

This study used the noon photos (5 photos) taken on June 4, 2019 with a focal length of 85 mm. The resulting point cloud had an average point-spacing of 1.5 cm and can be seen in Fig. 4c. We also use a TP dataset collected on March 19 2018, which is used as a reference for rock-fall detection. These TP point clouds (2018 and 2019) are created using the same 13 GCPs that were used in the sUAS datasets acquired in 2019. In addition, the TP models use camera self-calibrations generated using these 13 GCPs. This was done for consistency and to reduce differences in geo-referencing and camera self-calibration in the TP and sUAS point clouds from the use of a different number of GCPs.

## 4. Experimental Results

*4.1 Dataset Comparison and Edge Detection Parameters*

This section demonstrates the effect of the $\alpha$-molecules parameters on edge detection for the present datasets. We use the 2019 datasets, specifically the TP dataset (with 13 GCPs), the TLS dataset (georeferenced using targets), and the sUAS dataset (with GNSS + 13 GCPs). Table 1 shows their statistical comparison using the M3C2 algorithm. The RMSE comparison verifies cm-level agreements between the three datasets, thus, identified edges should exist in the same location (within few cm) for all datasets. Note that in Kromer et al. (2019), who used images from the same TP system and TLS datasets (using a Faro Focus x330), lower RMSE values are found for the TP-TLS comparison (approximately 1.1 cm) because in Kromer et al. (2019), comparisons are constrained to the lower rock area, while in this study a larger region is considered (Fig. 5).

Table 1. RMSE of M3C2 algorithm comparisons for the three multi-platform datasets. All datasets are georeferenced.

| | TP (13 GCPs) | TLS | sUAS (GNSS + 13 GCPs) |
|---|---|---|---|
| TP (13 GCPs) | - | 3.2 cm | 2.1 cm |
| TLS | 3.2 cm | - | 3.3 cm |
| sUAS (GNSS + 13 GCPs) | 2.1 cm | 3.3 cm | - |

Assessment of edge detection and search of input parameters uses the PFOM metric. Based on the search/test parameters identified for this test site (see section 2.1), we computed correlation coefficients of the tested parameter values with the resulting PFOM values for each scenario (i.e., TLS vs TP, TP vs sUAS, and TLS vs sUAS). This identifies which parameters have the most influence on the PFOM metric and thus on edge detection and helps us understand how different input parameter values change the resulting PFOM values. The strongest correlation in all three comparisons was found to be the maximum feature width parameter (correlation coefficients of -0.81, -0.60, and -0.72, respectively), while the second strongest was the minimum contrast threshold (correlation coefficients of -0.32, -0.30, and -0.40, respectively). Note that these correlations depend on the selected parameters tested in this paper. Figure 6 shows the PFOM variability based on the max feature width parameter, thus highlighting that highest PFOM values are retrieved with a 4 m maximum width and maximum length.

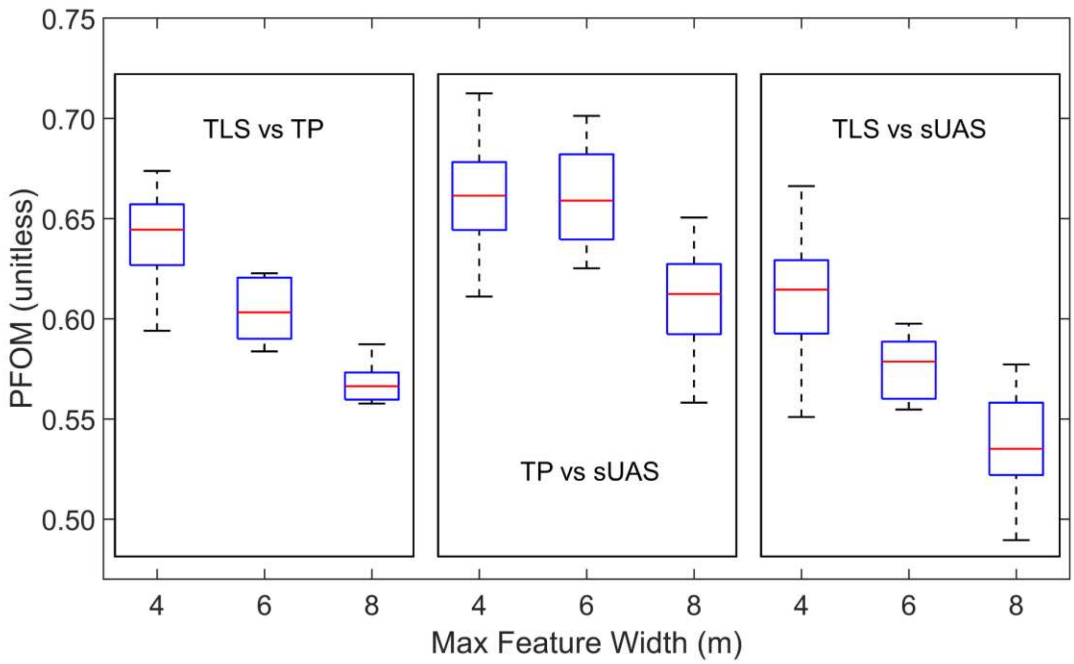



**Fig. 6.** variability of PFOM values based on max feature width.

Table 2 gives the parameters with the highest PFOM values for each comparison. The three comparisons yielded similar "best" parameters with minor differences in the number of orientations and minimum feature width for the TP-sUAS comparison. The TLS-sUAS and TLS-TP comparisons yielded the same parameters. The differences in PFOM values for the identified parameters in Table 2 are small. For example, the PFOM value of TP-sUAS using its "best" parameters is 0.712, while when we use the "best" parameters from the TLS-sUAS and TLS-TP comparisons, the PFOM value is 0.709. Therefore, for consistency we use the "best" parameters obtained from the TLS-sUAS and TLS-TP comparisons. Using these parameters, the PFOM values are 0.674 for TLS-TP and 0.666 for TLS-sUAS comparisons.

Table 2. $\alpha$-molecule parameters with the highest PFOM values for each comparison

| | Parameters that yield highest PFOM values | | |
|---|---|---|---|
| Parameter | TLS vs TP | TLS vs sUAS | TP vs sUAS |
| Scales per octave | 4 | 4 | 4 |
| Min contrast | 1 | 1 | 1 |
| Number of orientations | 2^8 | 2^8 | 2^12 |
| Max feature width (m) | 4 | 4 | 4 |
| Max feature length (m) | 4 | 4 | 4 |
| Min feature width (m) | 1 | 1 | 0.5 |
| Orientation operator | Rotation | Rotation | Rotation |

Figure 7a,b,c shows the identified edges for each comparison using the parameters of Table 2. Although there are some differences in detected edges between the three comparisons (for example see Fig. 7a versus Fig. 7b, 7c), what is important for the registration algorithm is that similar edges are identified (i.e., relative similarity). With respect to the latter, Fig. 7 shows very good agreement, as there are several edges that are identified in the compared datasets. Detected edges are different (in an absolute sense) because of the differences in gridding, as we grid datasets with grid extents based on the coverage of the reference dataset. For example, Figures 7a and 7c have different grids and grid extents, because the coverage of the two reference datasets is different In Figure 7a the grid is based on the extent of the TLS dataset and in Figure 7c the grid is based on the extent of the TP dataset. However, as mentioned above, relative similarity is assessed by the algorithm, and satisfactory results are therefore achieved in Figure 7 for the difference scenarios.

For comparison, we show the detected edges in the TLS versus sUAS datasets when 8 m are used for max feature width and length (Figure 7d), and 2 m for max feature length and width (Figure 7e). In addition, we show the detected edges in the TLS versus TP comparison using 8 m for max feature width and length (Figure 7f). When the max feature width and length parameters are increased, less edges are found (Figures 7d and 7f versus Figures 7b and 7c, respectively). When these parameters are decreased, more edges are detected (Figure 7e versus 7b); however, we should note that the edges with the 2 m max feature width and length are more fragmented, which cannot be seen in the figure. In addition, there are more mismatches and poorly shaped edges in the top area of Figure 7e where vegetation exists than in Figure 7b, which uses a 4 m max feature width and length.

To show the effect of the different edge detection results on registration, we created three scenarios. In all scenarios the reference dataset is the TLS, and the source dataset is the sUAS GNSS with 0 GCPs. This means that no GCPs are used, and geo-referencing is achieved from the GNSS image positions. The absence of GCPs can lead to poor camera self-calibration (Bolkas 2019); therefore, to remove such effects, we use the camera self-calibration derived from the case with 13 GCPs. Use of the 13 GCP self-calibration parameters in the sUAS dataset is denoted here as "sUAS (GNSS + 0 GCPs; Pre-calibration)". In the first scenario we use 4 m max feature width and length (derived from Table 2), in the second scenario we use 2 m max feature width and length, and in the third scenario we use 8 m max feature width and length. Note that the RMSE for comparison between the TLS and the sUAS (GNSS + 0 GCPs; Pre-calibration) is 42.2 cm. Since, we are using an accurate pre-calibration, the high RMSE indicates high georeferencing error due to the absence of GCPs. Using the first scenario the registration RMSE is 3.5 cm, using the second scenario the RMSE is 3.7 cm, and using the third scenario the RMSE is 4.4 cm. This result indicates that even when the input parameters are not the "best" and few edges are detected, the algorithm manages to register the two datasets with high accuracy, but with less accuracy than when using the optimal parameters. The following section deals with further assessment of the developed registration algorithm.

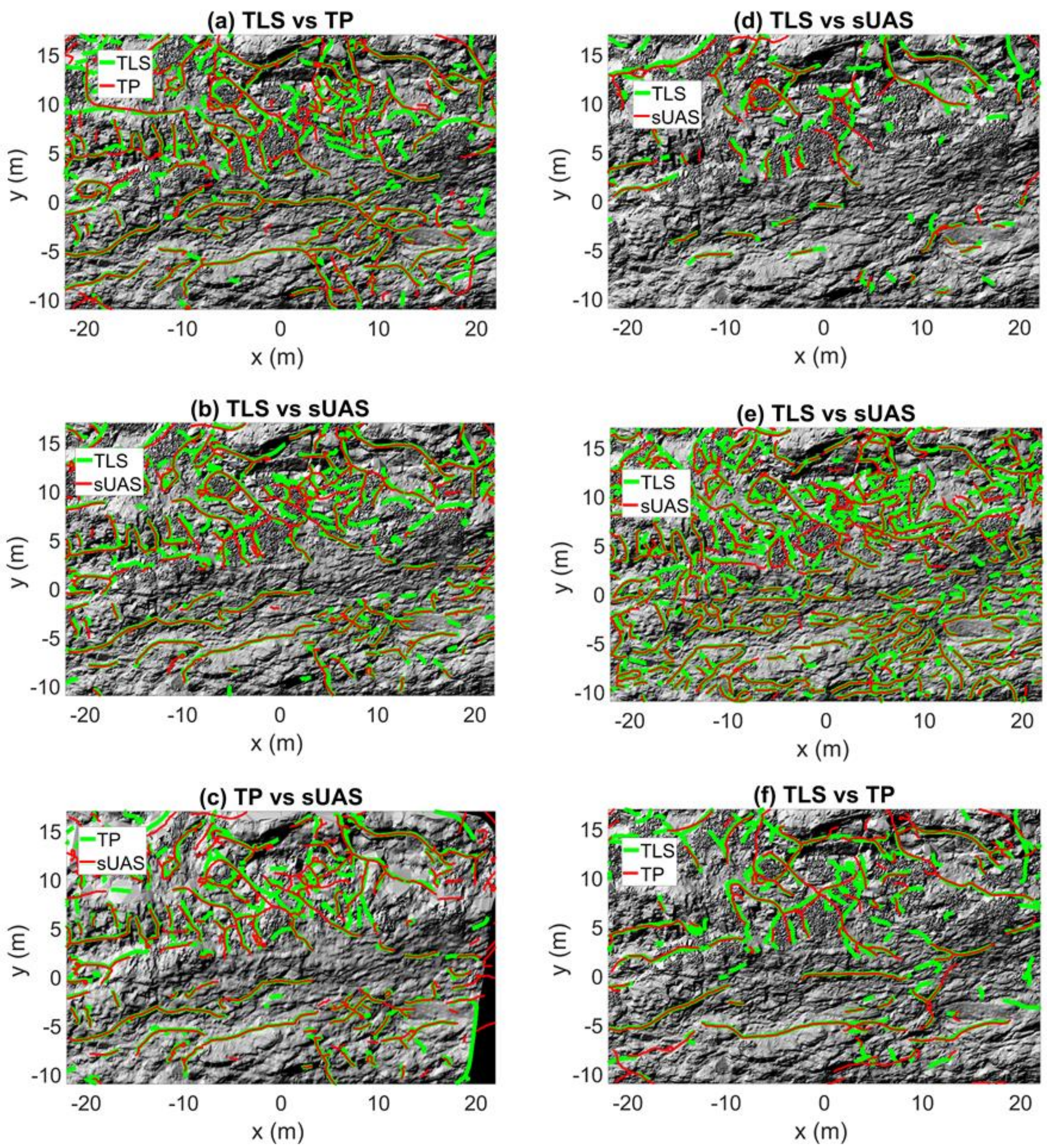



**Fig. 7.** Detected edges for each comparison based on different $\alpha$-molecule parameters. (a) TLS versus TP with parameters from Table 2; (b) TLS versus sUAS with parameters from Table 2; (c) TP versus sUAS with parameters from Table 2; (d) TLS versus sUAS with 8 m max feature width and 8 m max feature length; (e) TLS versus sUAS with 2 m max feature width and 2 m max feature length; (f) TLS versus TP with 8 m max feature width and 8 m max feature length. The sub-figures also show the dataset used as reference in the background. Note that reference edges (green) are shown thicker than source ones (red) to enhance illustration.

### *4.2 Effect of number of edges in the edge matching process*

To evaluate the effect of the DFD threshold on the edge matching, we counted the number of correctly and incorrectly matched edges through visual inspection. The results are plotted in Fig.

8 as a function of the DFD threshold value for each comparison (TLS vs TP, TLS vs sUAS, and TP vs sUAS). Note that the DFD was normalized using the maximum and minimum DFD values for each comparison, and thus is unitless. Figures 8a,b,c show that as the threshold increases, so do the number of correctly and incorrectly matched edges; therefore, the user has to decide the appropriate threshold through trial and error. We also calculated the ratio of the number of incorrectly matched edges over the number of correctly matched edges (Fig. 8d). A ratio value of 0.5 means that the number of incorrect matched edges is half the number of the correct matched edges. To reduce the effect of incorrect edges, this ratio value should remain low. We consider that ratio values of >0.3-0.4 are high and can introduce outliers in the Procrustes analysis. Note that many of the incorrect edges are found in vegetation areas, as vegetation is captured differently by the different sensors (see top areas in the sub-figures of Figure 7). Although such edges may be in the same location in the various datasets, their shape will be different, as they are affected differently by vegetation. Many of those "incorrect" edges were matched in the correct location; therefore, edge matching was successful, but edges with distinctly different shapes are labeled "incorrect", as they will have a negative impact (outliers) on the Procrustes analysis.

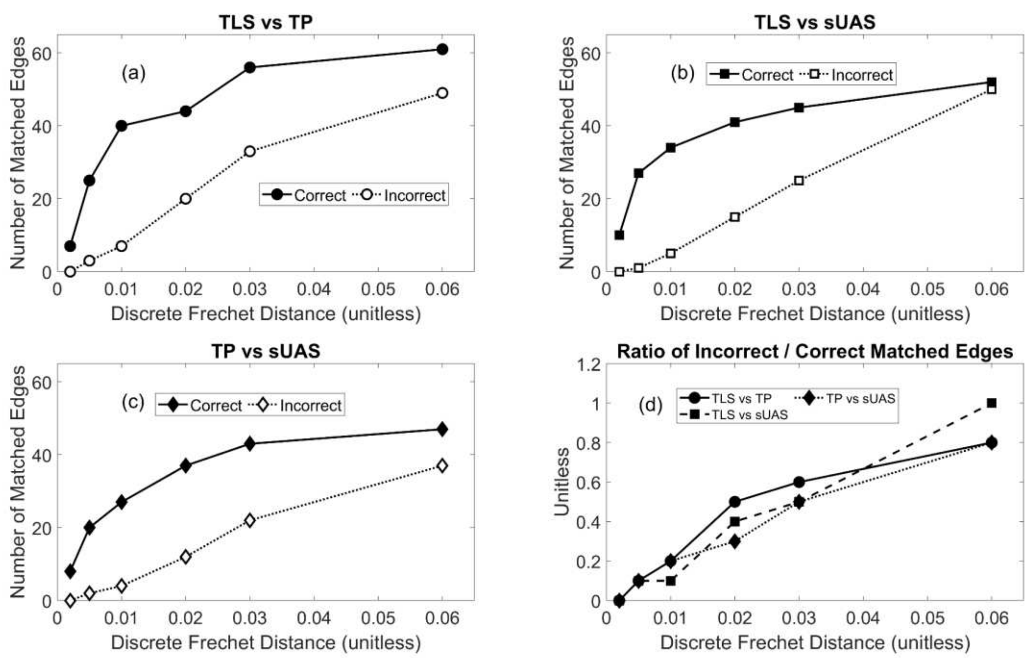



**Fig. 8.** Effect of DFD threshold on edge matching (a) TLS vs TP; (b) TLS vs sUAS; (c) TP vs sUAS; (d) ratio of the number of incorrect matched edges over the number of correct matched edges for each comparison. Note that the Discrete Fréchet Distance has been normalized to the maximum and minimum distances in each comparison, such that it is unitless.

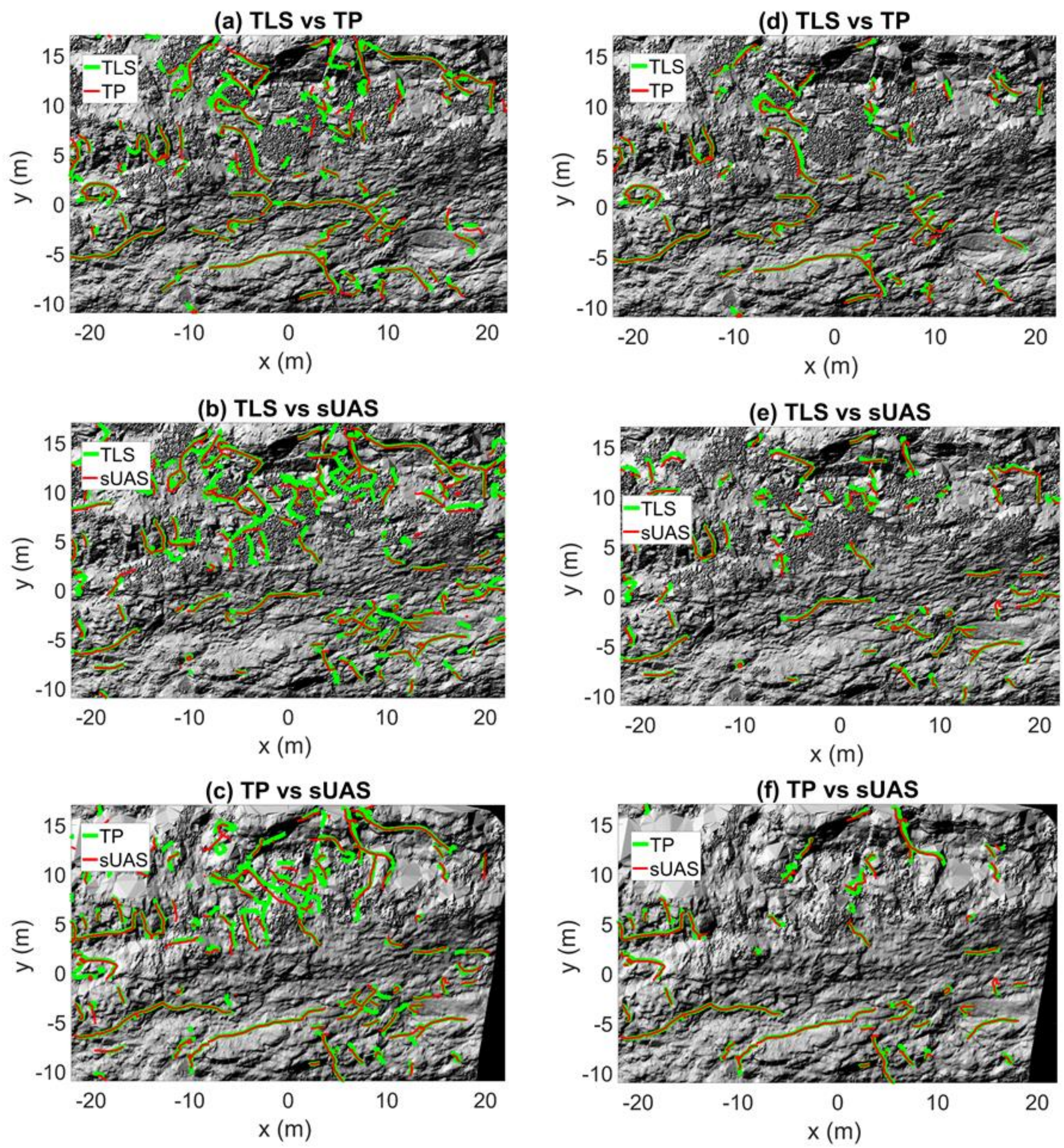



**Fig. 9.** Example result of the edge matching step using the discrete Fréchet Distance. (a) TLS vs TP; (b) TLS vs sUAS; (c) TP vs sUAS. Sub-figures (a), (b), and (c) use DFD threshold of 0.025. (d) TLS vs TP; (e) TLS vs sUAS; (f) TP vs sUAS. Sub-figures (d), (e), and (f) use DFD threshold of 0.01. The sub-figures also show the dataset used as reference in the background. Note that reference edges (green) are shown thicker than source ones (red) to enhance illustration.

Figure 9 shows an example of the edge matching step. The figure was created after computing the DFD between identified edges for each comparison. Figures 9a,b,c are created using a threshold of 0.025, while Figures 9d,e,f are created using a threshold of 0.01. These thresholds were used to show some of the edges with different shapes, found in the same location, in the top parts of the dataset area. As mentioned earlier, these are affected mostly by vegetation (Figures 9a,b,c). The influence of those edges is minimized when more strict thresholds are used (Figures 9d,e,f).

Furthermore, in the case with strict thresholding, edges with poor shape correspondence are typically of short length (1-2 meters), meaning that only few points will be matched and will therefore not have a significant impact on the Procrustes analysis and registration. The figure shows good edge matching performance overall. Furthermore, the figure shows that vegetation or data gaps do not adversely affect the edge matching process. This is further discussed in the following section (Section 4.3) where poor mismatches and outliers are removed in the Procrustes step.

### *4.3 Effect of grid-spacing and number of matched points on registration result*

This section evaluates the effect of grid-spacing on the registration accuracy. We test scenarios with grid-spacings of 10 cm, 5 cm, and 2.5 cm. The comparisons used are: (i) TP versus sUAS (GNSS + 0 GCPs; Pre-calibration); (ii) TP versus sUAS GNSS + 13 GCPs; (iii) TLS versus sUAS (GNSS + 0 GCPs; Pre-calibration); (iv) TLS versus sUAS GNSS + 13 GCPs. Table 3 shows the RMSE results for each grid-spacing scenario. In general, there is small effect of the selected grid on the registration result. The case with 10 cm grid spacing yields the highest RMSE values, but these are still within few mm of the RMSE values obtained using a 2.5 cm grid spacing. In general, small differences are found between the 5 cm and 2.5 cm grid-spacings. The 2.5 cm grid-spacing is preferred here because it matches the average grid-spacing of the input datasets. In the presence of significant scale differences, the source point cloud is scaled with an initial scale value (see section 2.1), then the algorithm uses a grid step size based on the average point density of the point cloud, and the process is refined through iteration (see scenarios in section 4.5).

Table 3. Effect of grid-spacing on registration; values are RMSE in cm.

| | | | Proposed Algorithm: Grid-spacing | | |
|---|---|---|---|---|---|
| Comparison | Initial | ICP | 10 cm | 5 cm | 2.5 cm |
| TP (13 GCPs) vs sUAS (GNSS + 0 GCPs; pre-calibration) | 35.5 | 3.3 | 3.0 | 3.1 | 2.9 |
| TP (13 GCPs) vs sUAS (GNSS + 13 GCPs) | 2.1 | 2.0 | 2.4 | 2.4 | 2.0 |
| TLS vs sUAS (GNSS + 0 GCPs; pre-calibration) | 42.2 | 3.9 | 4.1 | 3.8 | 3.5 |
| TLS vs sUAS (GNSS + 13 GCPs) | 3.3 | 2.6 | 2.7 | 2.6 | 2.5 |

For the same dataset comparisons, we provide the point correspondences that were identified when using the 2.5 cm grid spacing (Fig. 10). The number of identified points is about 3,500 in all cases, and points have been identified in several parts of the study area. This is an adequate number of points for the Procrustes analysis. Figure 10 also shows how poorly shaped and poorly matched edges in the top part of the study areas, because of vegetation, have been removed from the Procrustes analysis, thus minimizing their effect.

Figure 11 shows the effect on registration RMSE when we randomly reduce the number of points. In most cases, we see small differences even when only a few hundred points are used. Larger RMSE values are found when the number of identified points drops below one hundred, as the network geometry of matched points can deteriorate, and the influence of poorly matched points can be increased. To provide additional insights and to show how the number of correct / incorrect edges affect the registration we have considered additional analysis scenarios using the TP (13 GCPs) versus sUAS (GNSS+ 0 GCPs; pre-calibration) comparison. Using 25 correct edges and 5 incorrect with about 2,000 matched points, the resulting RMSE is 2.9 cm. Using three manually selected edges (one at the top right corner and two at the bottom of the site) with 70 matched points, we were able to register the two point clouds with an RMSE of 3.3 cm. When using about 28 correct edges and 20 incorrect ones with about 2,000 matched points, we obtained an RMSE of 6.1 cm. While the ideal scenario is to have many correct edges and few or no incorrect ones, accurate registration is possible even with just a few "correct" edges. In the case with many correct and incorrect edges, there is a higher level of contamination from poorly shaped or poorly matched edges. In this paper, we have demonstrated that the level of contamination can depend on the selected parameters for edge detection, edge matching, and identification of point correspondences. As shown in Figure 10, after thresholding to remove poor point correspondences in the Procrustes step, the level of contamination is low. In addition, the user, by selecting appropriate parameters, can control and reduce this level of contamination. Robust outlier removal methods (e.g., Bustos and Chin 2017; Li et al. 2020a) could also be utilized in the future as necessary.

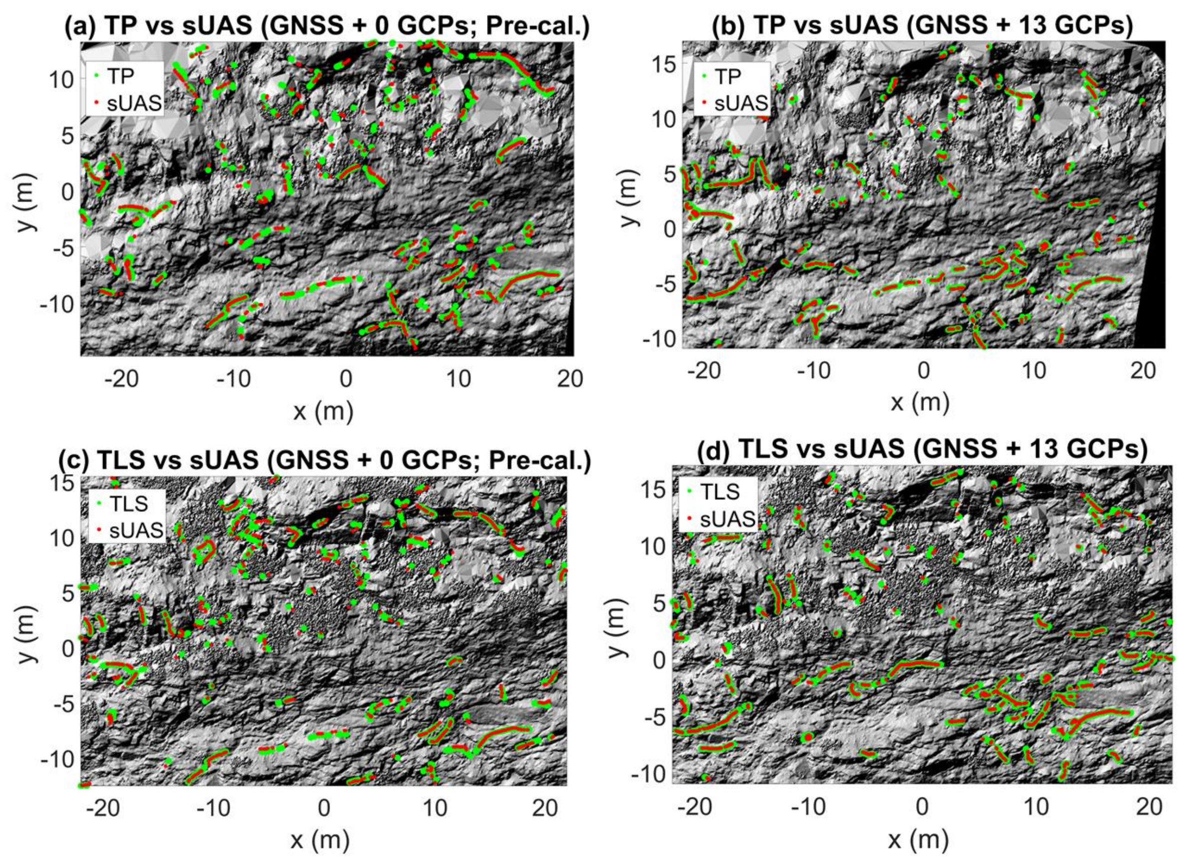


**Fig. 10.** Examples of identified matched points (a) TP (13 GCPs) versus sUAS (GNSS + 0 GCPs; pre-calibration); (b) TP (13 GCPs) versus sUAS (GNSS + 13 GCPs) (c) TLS versus sUAS (GNSS + 0 GCPs; pre-calibration) (d) TLS versus sUAS (GNSS + 13 GCPs). Note that reference points (green) are shown two times larger than source ones (red) to enhance illustration.

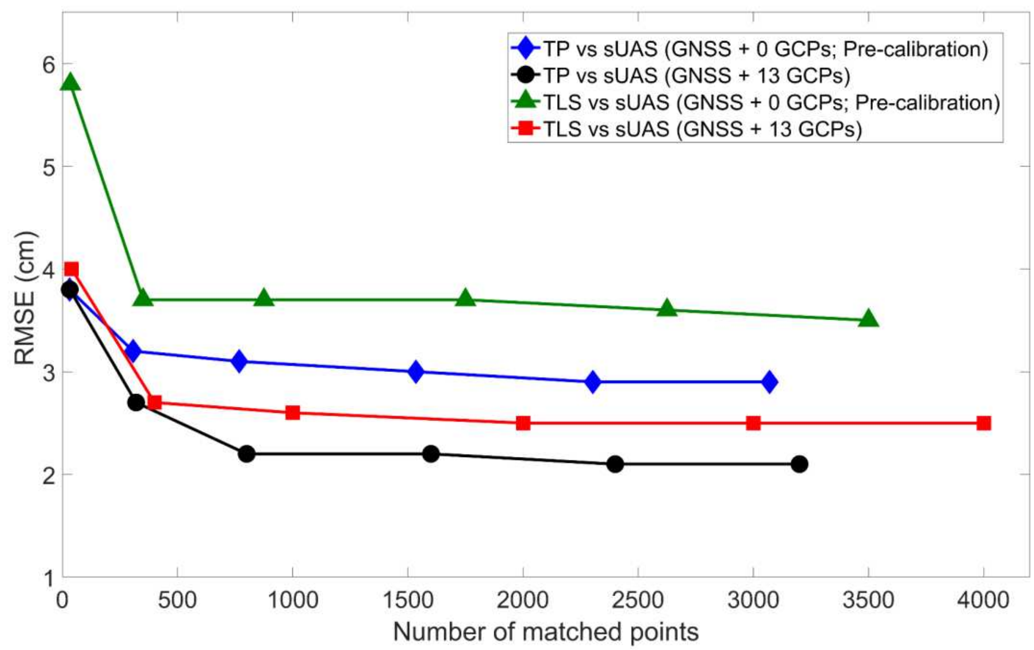
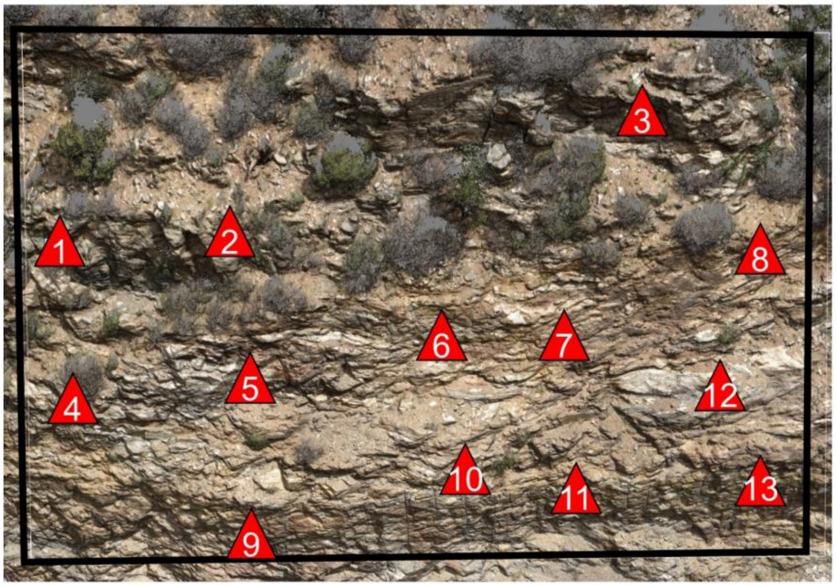



**Fig. 11.** Registration RMSE as a function of identified matched points.

*4.4 Registration with varying numbers of GCPs*

In this section, we apply the developed algorithm in different GCP scenarios. The scenarios here use the TP and TLS datasets as reference and we vary the number of GCPs used in the sUAS dataset, which affects its georeferencing. Then the proposed algorithm is applied to show the cases where it can contribute to enhancing registration of multi-platform point clouds. Scenarios are provided with sUAS point clouds that are generated with accurate GNSS positioning for the aerial images. Finally, for comparison we have applied the ICP algorithm for each GCP scenario. Figure 12 shows the GCP scenario that is tested here and how we go from the zero GCP scenario (GNSS-assisted orientation) to 13 GCPs. As mentioned in the previous section, having no GCPs affects camera self-calibration in sUAS-generated point clouds. Therefore, we present scenarios where camera self-calibration relies on identified keypoints on the images and using the self-calibration that is derived from the GNSS + 13 GCP case. The latter is denoted as "pre-calibration", thus simulating the case where an accurate pre-calibration is available.




| Scenario | GCP ID |
|---|---|
| 0 GCPs | - |
| 1 GCPs | 6 |
| 2 GCPs | 3, 9 |
| 3 GCPs | 1, 3, 9 |
| 4 GCPs | 1, 3, 9, 13 |
| 5 GCPs | 1, 3, 6, 9, 13 |
| 6 GCPs | 1, 3, 6, 8, 9, 13 |
| 13 GCPs | All |

**Fig. 12.** Georeferencing scenarios with GNSS-assisted orientation and varying GCP number.

Table 4 shows the results of the sUAS comparisons using the TP dataset as reference and Table 5 shows the results of the sUAS comparisons using the TLS dataset as reference. Tables 4 and 5 show the RMSE values before the application of the proposed registration algorithm. For the case with 0 GCPs (GNSS-assisted orientation) these are 44.0 cm and 50.5 cm for the TP and TLS datasets as reference, respectively. The main reasons for the high RMSE error are poor georeferencing and camera self-calibration due to the absence of GCPs. Using the camera calibration parameters of the case with 13 GCPs, these drop to 35.3 cm and 42.2 cm for the TP and TLS datasets as reference, respectively. This indicates that most of the error originates from poor georeferencing. After registration with the proposed algorithm the RMSE drops to 5.8 cm and 6.5 cm for the TP and TLS reference datasets respectively, without good camera pre-calibration, and 2.9 cm and 3.5 cm when we use the camera calibration from the 13 GCP case. These results highlight that the proposed algorithm can be used to support rockfall monitoring and reduce the number of GCPs that are required for georeferencing or registration, when a good camera pre-calibration exists. Initial RMSE values using 1 GCP are at the 20 cm – 30 cm level. After registration with the proposed algorithm these drop to 2.8 cm – 3.0 cm, when using camera pre-calibration. For cases with more GCPs there is little or no improvement, as the GNSS and GCPs manage to georeference the various point clouds. For the comparisons with the TLS as reference, there is improvement in all GCP cases, because there is an initial mean bias of about 1.5 cm that is removed after registration with the proposed algorithm. The source of this mis-registration is believed to be due to the different geo-referencing approach of the laser scanning and photogrammetric point clouds. Recall from Fig. 5 that the TLS dataset uses the control points on the opposite side of the highway, while the TP and sUAS datasets use the GCPs on the rock-surface.

Comparison with the ICP algorithm as implemented in Cloud Compare (CloudCompare 2015) in Tables 4 and 5 shows that the proposed algorithm generates comparable results, and in many cases, better results are found. The cases where the proposed algorithm provides better results than the ICP are mostly in the 0 and 1 GCPs scenarios. The ICP algorithm sometimes can be trapped in local minima when initial registration is poor (Attia and Slama 2017), meaning that the proposed algorithm provides the best relative performance in such situations. The RMSE values for the 0 and 1 GCP scenarios before registration are about 30-50 cm, which indicate a poorer initial registration than the remaining scenarios. Figure 13 shows the histograms for the TP versus sUAS with GNSS using 0 GCPs and 1 GCPs cases (both using the camera calibration from the 13 GCP case). The histograms of the M3C2 distance when using the proposed algorithm show higher peaks and smaller tails than the ICP algorithm. This highlights the improvement after the application of the proposed algorithm, as well as the better performance compared to the ICP algorithm. In addition, they validate that the proposed algorithm can be used to support the registration of multiplatform point clouds for rockfall monitoring.

Table 4. Registration of the sUAS dataset to the TP dataset for various GCP scenarios. RMSE values are shown.

| GCP number | Before (cm) | Proposed Algorithm (cm) | ICP (cm) |
|---|---|---|---|
| 0 | 44.0 | 5.8 | 6.1 |
| 0 (pre-calibration) | 35.5 | 2.9 | 3.3 |
| 1 | 30.0 | 3.5 | 3.5 |
| 1 (pre-calibration) | 23.6 | 2.8 | 3.0 |
| 2 | 2.3 | 2.0 | 2.1 |
| 3 | 2.2 | 2.1 | 2.1 |
| 4 | 2.1 | 2.1 | 2.1 |
| 5 | 2.0 | 2.0 | 2.0 |
| 6 | 2.1 | 2.0 | 2.0 |
| 13 | 2.1 | 2.0 | 2.0 |

Table 5. Registration of the sUAS dataset to the TLS dataset for various GCP scenarios. RMSE values are shown.

| GCP number | Before (cm) | Proposed Algorithm (cm) | ICP (cm) |
|---|---|---|---|
| 0 | 50.5 | 6.5 | 8.5 |
| 0 (pre-calibration) | 42.2 | 3.5 | 3.9 |
| 1 | 35.6 | 3.6 | 5.7 |
| 1 (pre-calibration) | 28.2 | 3.0 | 3.8 |
| 2 | 4.0 | 2.6 | 2.8 |
| 3 | 4.2 | 2.5 | 2.6 |
| 4 | 3.4 | 2.6 | 2.5 |
| 5 | 3.2 | 2.6 | 2.5 |
| 6 | 3.2 | 2.6 | 2.6 |
| 13 | 3.3 | 2.5 | 2.6 |

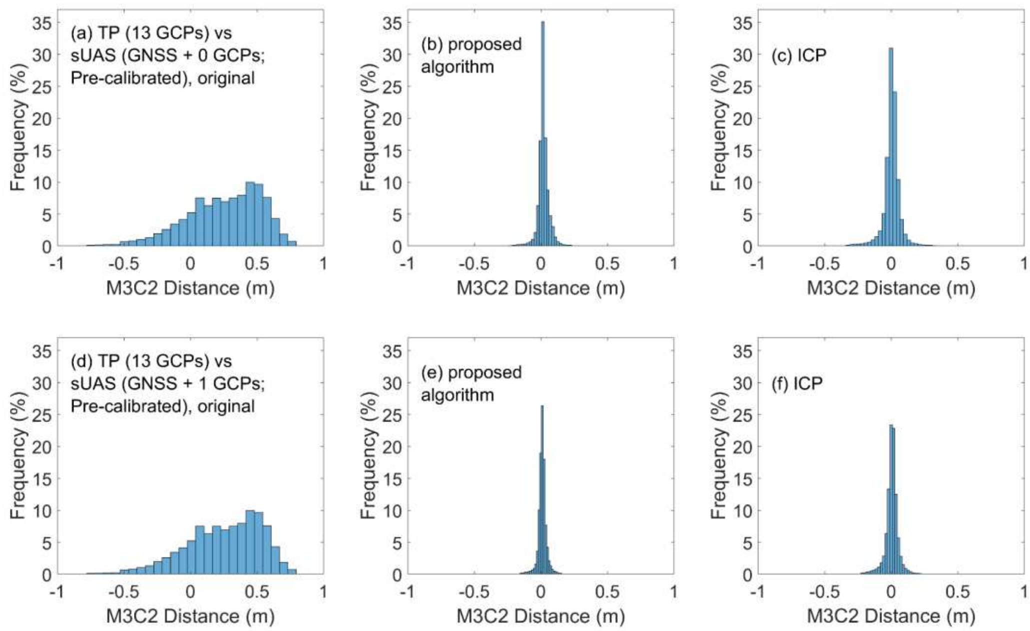



**Fig. 13.** Histograms for the TP (13 GCPs) versus sUAS (GNSS + 0 GCPs; Pre-calibrated) and TP (13 GCPs) versus sUAS (GNSS + 1 GCPs; Pre-calibrated). Differences are calculated using the M3C2 distance. (a) TP (13 GCPs) versus sUAS (GNSS + 0 GCPs; Pre-calibrated), original; (b) TP (13 GCPs; Pre-calibrated) versus sUAS (GNSS + 0 GCPs; Pre-calibrated), proposed algorithm; (c) TP (13 GCPs) versus sUAS (GNSS + 0 GCPs; Pre-calibrated), ICP; (d) TP (13 GCPs) versus sUAS (GNSS + 1 GCPs; Pre-calibrated), original; (e) TP (13 GCPs) versus sUAS (GNSS + 1 GCPs; Pre-calibrated), proposed algorithm; (f) TP (13 GCPs) versus sUAS (GNSS + 1 GCPs; Pre-calibrated), ICP.

*4.5 Registration scenarios with translations, rotations, and scale*

Registration results for multi-platform datasets with significant translations, rotations, and scale are provided in this section, for further evaluation of the proposed algorithm. Scenarios considered in this section resemble cases where the two datasets have significantly different coordinate systems (perhaps because the number of GCPs is insufficient for georeferencing or a different coordinate system is used). Five scenarios are investigated in this section (see Table 6). The scenarios include translations only in all coordinate-directions, translations and rotations in all coordinate-directions, and translations, rotations and scale differences in all coordinate-directions. These scenarios present significant translations, rotations, and even differences in scale (translations of 500 m, rotations of 30° and 45°, scale differences of 0.5 and 0.8). The scenarios are applied to the sUAS (GNSS + 13 GCPs) dataset using the TP (13 GCPs) as reference and to the TP (13 GCPs) dataset using the TLS dataset as reference. It is worth noting that using the PCA and PCT, we manage to approximately register the input point clouds with an RMSE of about 0.6 m to 0.7 m for the simulated cases with translations and rotations. When scale is added, the approximate registration is about 0.7 m to 1.1 m. This shows a sufficient approximate alignment for the datasets of this site, which is important for the subsequent steps of the registration algorithm. The level of accuracy required for the initial registration is related to the spacing between detected edges. The detected edges in Figure 7 are several meters apart; therefore, a meter-level initial registration error is required for this site. The adequacy of the initial registration is also demonstrated by the results of the edge matching step (Figure 9), as an insufficient initial registration would produce a poor edge matching.

The TP versus sUAS dataset comparison has an original deviation of 2.1 cm; in all scenarios tested the proposed algorithm manages to register the two point clouds with an RMSE of 2.0 cm to 2.3 cm. Note that for the scenarios with scale, the proposed algorithm was iterated four to five times. For the TLS and TP datasets, their original comparison of the resulted in an RMSE of 3.2 cm. The proposed algorithm registered the two point clouds with an RMSE of 2.5 – 2.9 cm, in all scenarios tested. Scenarios with scale were also iterated four to five times in this case. In the presence of scale in the source dataset, an initial scale value is estimated using the ratio of the singular values between the two datasets (see section 2.1). The algorithm then selects the grid step for the source dataset based on its average point density, as the algorithm can support different grid steps for the target and source datasets. Then the scale, grid step, and edge detection input parameters (maximum feature length and width, and minimum feature width) are refined in each iteration.

Table 6. RMSE results of translations, rotations, and scale scenarios for the TP (13 GCPs) versus sUAS (GNSS + 13 GCPs) and TLS versus TP datasets.

| Comparisons | Scenarios | RMSE (cm) |
|---|---|---|
| TP (13 GCPs) versus sUAS (GNSS + 13 GCPs) | Original comparison | 2.1 |
| | 500-m offset | 2.0 |
| | 500-m offset and 30° rotation | 2.0 |
| | 500-m offset and 45° rotation | 2.0 |
| | 500-m offset, 45° rotation, and 0.8 scale | 2.1 |
| | 500-m offset, 45° rotation, and 0.5 scale | 2.3 |
| TLS versus TP (13 GCPs) | Original comparison | 3.2 |
| | 500-m offset | 2.5 |
| | 500-m offset and 30° rotation | 2.6 |
| | 500-m offset and 45° rotation | 2.6 |
| | 500-m offset, 45° rotation, and 0.8 scale | 2.8 |
| | 500-m offset, 45° rotation, and 0.5 scale | 2.9 |
| TLS versus TP (no GCPs; with pre-calibration) | ICP | 2.2 |
| | Proposed algorithm | 2.2 |

To create a more realistic scenario, we created a TP point cloud without using any GCPs and using the camera calibration of the 13 GCPs case. Therefore, this point cloud is lacking scale and coordinate system definition. After approximately scaling the two datasets using their coverage, the ICP managed to register the two point clouds with an RMSE of 2.2 cm. Using the same approximately scaled datasets as input, the proposed algorithm managed to register the two point clouds with the same RMSE of 2.2 cm. These results demonstrate that the proposed algorithm is capable of registering point clouds with significant differences in position, orientation, and scale.

### *4.6 Comparison in two epochs*

Placing and maintaining GCPs on rock formations is both expensive and difficult, as GCPs will get destroyed or deteriorate over time. This section demonstrates how the proposed algorithm can be used to address such issues and recover the registration between datasets collected in two different epochs. In these scenarios, we attempt to register datasets collected in 2018 and 2019 in the same rock formation. The 2018 datasets we use are the TP dataset with 13 GCPs and the TLS dataset, which is not georeferenced to NAD 83 (2011), meaning it uses a local (unknown) coordinate system. The 2019 dataset that we use is the TP dataset, but no GCPs have been used, assuming that all GCPs were destroyed. Due to the hypothetical absence of GCPs, we used the camera calibration from 2018. Note that when using the GCPs, the TP (2018) versus TP (2019) comparison yields a 1.5-cm RMSE, so this is the target RMSE for the algorithm. We also use the sUAS dataset with GNSS + 0 GCPs with camera calibration from the 13 GCP case, which provides a georeferencing of 35.2 cm with respect to the TP (2018) dataset.

Table 7 shows the results of the different scenarios. The table gives the RMSE values after the registration with the proposed algorithm and registration using the ICP algorithm. Note that to reduce the effect of rockfalls in the calculation of the RMSE values, we consider M3C2 distances that are within ±10 cm, as rockfalls in this study area are expected to create changes that are higher than that. In addition, in all cases, both the proposed algorithm and ICP are applied with scale. The scale parameter is important for the TP (2018) versus TP (2019) comparison, as the 2019 point cloud does not use any GCPs. We use as input the datasets that have been approximately scaled using their coverage.

The resulting RMSE is 1.8 cm, which is very close to the original RMSE of 1.5 cm and the RMSE of 1.6 cm achieved by the ICP. In the TLS (2018) versus TLS (2019) comparison, the proposed algorithm achieved an RMSE of 1.5 cm and the ICP an RMSE of 1.2 cm. In the TP (2018) versus sUAS (2019) and TLS (2018) versus sUAS (2019) comparisons, the proposed algorithm achieved an RMSE of 2.5 cm and 3.1 cm, respectively. The ICP algorithm achieved RMSE values of 3.3 cm and 3.5 cm, respectively. These results demonstrate that the proposed algorithm can be used for the registration of multi-platform datasets in two different epochs.

Table 7. Registration experiments of datasets in two different epochs, RMSE values in cm are shown.

| | Original (cm) | Proposed algorithm (cm) | ICP (cm) |
|---|---|---|---|
| TP (2018, 13 GCPs) vs TP (2019, 0 GCPs; with pre-calibration) | 1.5 | 1.8 | 1.6 |
| TLS (2018) vs TLS (2019) | Unknown | 1.5 | 1.2 |
| TP (2018, 13 GCPs) vs sUAS (2019; GNSS + 0 GCPs; Pre-calibration) | 35.2 | 2.5 | 3.3 |
| TLS (2018) vs sUAS (2019; GNSS + 0 GCPs; Pre-calibration) | Unknown | 3.1 | 3.5 |

Having registered datasets in two different epochs, we provide some examples of rockfalls detected in the data (Fig. 14). The site consistently presents rockfalls of small magnitude ranging in volume from 0.003 $m^3$ to 0.04 $m^3$ (Kromer et al. 2019), meaning detection at the 95% confidence level is challenging. Figure 13a shows rockfalls identified using the TP (2018, 13 GCPs) versus TP (2019, 13 GCPs) comparison, which had an RMSE of 1.5 cm. This case is used as a reference, as it has the lowest RMSE value. Figure 14b shows the results for the TLS (2018) versus TLS (2019) registered using the ICP algorithm. This case is used as reference for the TLS comparisons. Note that the TLS (2018) dataset had a smaller coverage than all other datasets, hence, a smaller area is shown in the Figures in the right side of Fig. 14. We limit rockfall detection on the lower part of the rock formation, as in the upper part differences due to vegetation and poor point cloud reconstruction (data gaps) might be present.

Figures 14c,d,e,f show cases where point clouds are registered using the proposed algorithm. Of note is that whether we use the TP dataset or the TLS dataset as reference, the same rockfalls are generally detected. A few discrepancies were found, as the 2018 TLS dataset was collected one month before the 2018 TP dataset (the reference TLS dataset was collected on February 13 2018 and the reference TP dataset was collected on March 19 2018). To highlight the similarity in rockfall detection, we have circled some rockfalls that are detected in all comparisons. Recall that the figures on the left side of Fig. 14 use the TP (2018) dataset as reference, while the right side figures use the TLS (2018) dataset as reference. The sUAS (2019) dataset manages to detect the same rockfalls as the TP and TLS datasets, although there are some differences related to the higher RMSE value (see Table 6). These comparisons demonstrate that the proposed algorithm can be used to support rockfall monitoring, especially if rockfalls are of larger magnitude than the ones encountered at the site considered in this study.

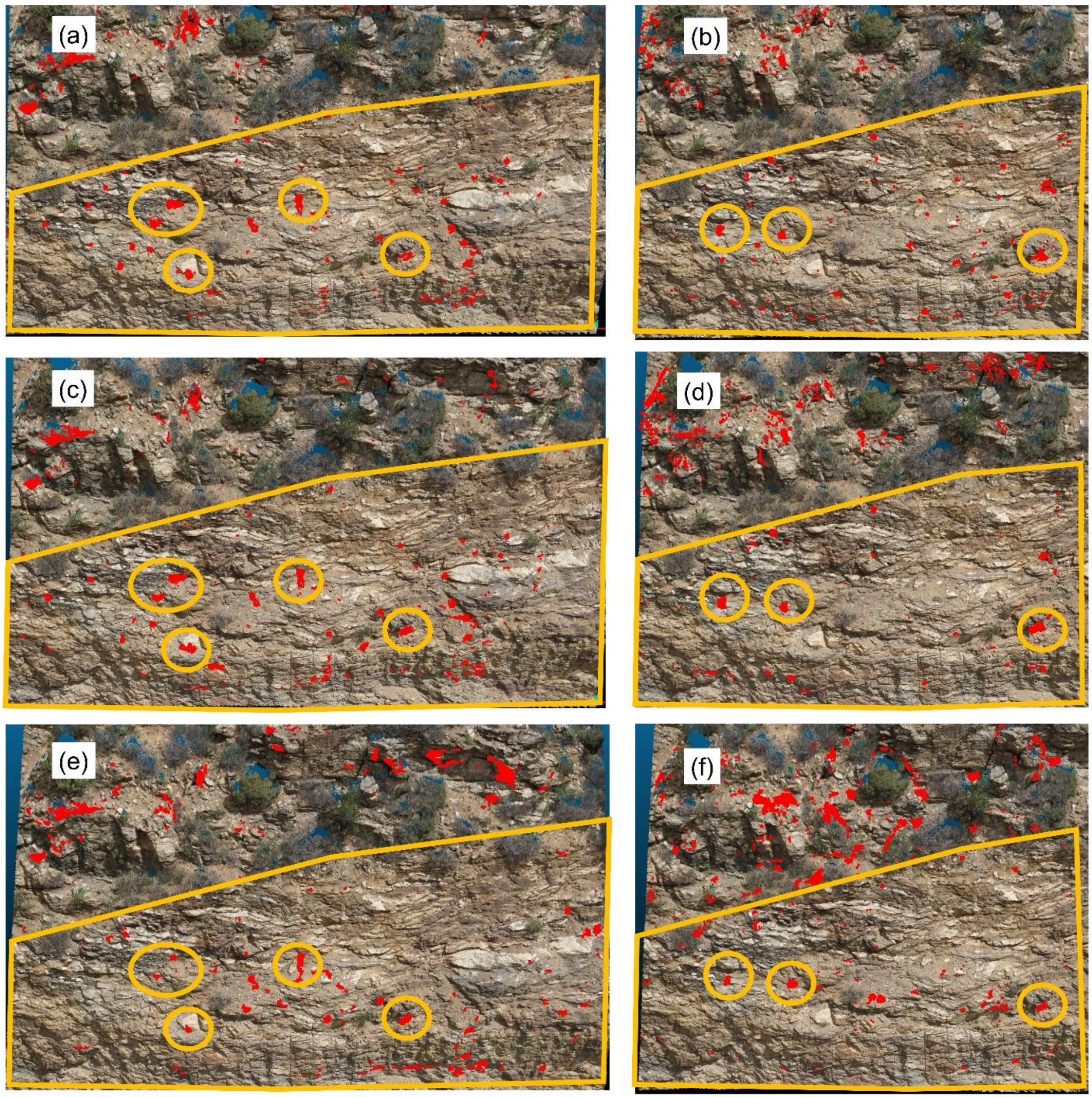



**Fig. 14.** Detected rock changes overlaid on the sUAS dataset (a) TP (2018, 13 GCPs) vs TP (2019, 13 GCPs) original comparison (b) TLS (2018) vs TLS (2019), ICP; (c) TP (2018, 13 GCPs) vs TP (2019, 0 GCPs; with pre-calibration), proposed algorithm; (d) TLS (2018) vs TLS (2019), proposed algorithm; (e) TP (2018, 13 GCPs) vs sUAS (2019; GNSS + 0 GCPs; Pre-calibration), proposed algorithm; (f) TLS (2018) vs sUAS (2019; GNSS + 0 GCPs; Pre-calibration), proposed algorithm.

## 5. Conclusions

Monitoring of rock surfaces relies on accurate registration of point cloud datasets. Sometimes these point clouds are generated from multi-platform sources such as TLS, TP, and sUAS (with and without GNSS). Registration typically relies on GCPs or other targets on the rock-surface. However, maintaining such GCPs is expensive and difficult. Registration of multi-platform datasets is challenging, as they may have different point-densities, data gaps/overlap, and/or quality, and surrounding characteristics of identified keypoints might be different in point clouds obtained from different methods. This paper presented a semi-automatic registration method that relies on edge detection of geological traces. The developed algorithm can register multiplatform and multiscale point clouds in complex rock surfaces. Edges are distinct features that can be matched in multi-platform point clouds. We have also provided several examples with scale difference between the two input point clouds, showing that the algorithm can be used in multiscale datasets. The algorithm utilized a recently developed multi-scale transform method to detect edges, termed $\alpha$-molecules, which offers a unified framework for multi-scale edge detection. Edge are matched automatically using the DFD. Matched points extracted from detected corresponding edges were then used for deriving translations, rotations, and scale (when necessary) through a Procrustes analysis. To assess the performance of the algorithm, we have utilized several multi-platform point clouds from an outcrop in Colorado. In addition, for comparison we have provided registration results using the ICP method.

The sUAS scenarios with varying numbers of GCPs showed that the proposed algorithm can support registration, especially when 0 and 1 GCPs are used as in such cases when geo-referencing using GNSS is at the few decimeter level. In the GNSS + 0 GCP and GNSS + 1 GCP cases, the proposed algorithm managed to register the sUAS point cloud to the TP and TLS point clouds with an RMSE of a few cm and, in general, performed better than the ICP. The ICP can get trapped in local minima in cases of insufficient initial registration, thus, the proposed algorithm can contribute in such cases. Scenarios with simulated translations, rotations, and scale showed that the proposed algorithm can register the two multi-platform point clouds with mm differences from their original RMSE value. Comparison of point clouds acquired in two different epochs demonstrated that the proposed algorithm can register multi-platform point clouds acquired in different epochs at the cm level and support detection of rockfall events. Despite the small rockfalls in the study area, the analysis showed promising and consistent results, which can further be improved with future evaluation, assessment, and development of the proposed algorithm. We plan to evaluate the algorithm in other study sites to further assess the performance of the developed algorithm and further explore its potential, especially in cases where rockfalls comprise a larger proportion of the scene. Further assessment will also help us identify generally reasonable input parameters. In addition, it is worth exploring merging the various datasets, exploiting their synergies to derive complete point clouds of the rock surface. The proposed registration algorithm would be important in that case to ensure accurate registration of the various datasets.

Compared to existing registration algorithms, that often rely on point correspondences, the proposed registration algorithm utilizes geological edges (discontinuity traces) that are important for quantitative analysis in rockmass hazard assessment. This presents the potential for creating a unified framework of edge utilization in the context of geotechnical/geological engineering.

**Acknowledgements**

The following surveying engineering students are acknowledged for their help during data collection: John Chapman, Vincent Pavill IV, Claudia Leu. The trip to Colorado was funded by Chancellor Endowment Funds from Penn State Wilkes-Barre.

**References**

Abellán, A., Calvet, J., Vilaplana, J. M., & Blanchard, J. (2010). Detection and spatial prediction of rockfalls by means of terrestrial laser scanner monitoring. *Geomorphology*, *119*(3-4), 162-171.

Abellán, A., Oppikofer, T., Jaboyedoff, M., Rosser, N. J., Lim, M., & Lato, M. J. (2014). Terrestrial laser scanning of rock slope instabilities. *Earth surface processes and landforms*, *39*(1), 80-97.

Agisoft LLC (2020) Agisoft Metashape User Manual Professional Edition, Version 1.6. https://www.agisoft.com/pdf/metashape-pro_1_6_en.pdf

Attia, M., & Slama, Y. (2017, July). Efficient initial guess determination based on 3d point cloud projection for icp algorithms. In *2017 International Conference on High Performance Computing & Simulation (HPCS)* (pp. 807-814). IEEE.

Bae, K. H., & Lichti, D. D. (2008). A method for automated registration of unorganised point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *63*(1), 36-54.

Besl, P. J., & McKay, N. D. (1992, April). Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures* (Vol. 1611, pp. 586-606). International Society for Optics and Photonics.

Bolkas, D. (2019). Assessment of GCP Number and Separation Distance for Small UAS Surveys with and without GNSS-PPK Positioning. *Journal of Surveying Engineering*, *145*(3), 04019007.

Bolkas, D., Naberezny, B., & Jacobson, M. G. (2021). Comparison of sUAS Photogrammetry and TLS for Detecting Changes in Soil Surface Elevations Following Deep Tillage. *Journal of Surveying Engineering*, *147*(2), 04021001.

Bolkas, D., & Martinez, A. (2018). Effect of target color and scanning geometry on terrestrial LiDAR point-cloud noise and plane fitting. *Journal of Applied Geodesy*, *12*(1), 109-127.

Bolkas, D., Sichler, T. J., & McMarlin, W. (2019). A case study on the accuracy assessment of a small UAS photogrammetric survey using terrestrial laser scanning. *Surveying and Land Information Science*, *78*(1), 31-44.

Bolkas, D., Vazaios, I., Peidou, A., & Vlachopoulos, N. (2018). Detection of rock discontinuity traces using terrestrial LiDAR data and space-frequency transforms. *Geotechnical and Geological Engineering*, *36*(3), 1745-1765.

Brenner, C., Dold, C., & Ripperda, N. (2008). Coarse orientation of terrestrial laser scans in urban environments. *ISPRS journal of photogrammetry and remote sensing*, *63*(1), 4-18.

Bustos, A. P., & Chin, T. J. (2017). Guaranteed outlier removal for point cloud registration with correspondences. *IEEE transactions on pattern analysis and machine intelligence*, *40*(12), 2868-2882.

Cai, Z., Chin, T. J., Bustos, A. P., & Schindler, K. (2019). Practical optimal registration of terrestrial LiDAR scan pairs. *ISPRS journal of photogrammetry and remote sensing*, *147*, 118-131.

Canny JF (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Machine Intell 8(6):34-43

Chen, Y., & Medioni, G. (1992). Object modelling by registration of multiple range images. *Image and vision computing*, *10*(3), 145-155.

Cheng, L., Chen, S., Liu, X., Xu, H., Wu, Y., Li, M., & Chen, Y. (2018). Registration of laser scanning point clouds: A review. *Sensors*, *18*(5), 1641.

Cheng, L., Tong, L., Li, M., & Liu, Y. (2013). Semi-automatic registration of airborne and terrestrial laser scanning data using building corner matching with boundaries as reliability check. *Remote Sensing*, *5*(12), 6260-6283.

Cheng, L., Tong, L., Wu, Y., Chen, Y., & Li, M. (2015). Shiftable leading point method for high accuracy registration of airborne and terrestrial LiDAR data. *Remote Sensing*, *7*(2), 1915-1936.

CloudCompare (2015) CloudCompare Version 2.6.1 User manual. http://www.cloudcompare.org/doc/qCC/CloudCompare%20v2.6.1%20-%20User%20manual.pdf

Da Cunha, A. L., & Do, M. N. (2005). Bi-orthogonal filter banks with directional vanishing moments [image representation applications]. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP'05). IEEE International Conference on* vol. 4, pp. iv-553.

Danziger, Z. (2021). Discrete Frechet Distance (https://www.mathworks.com/matlabcentral/fileexchange/31922-discrete-frechet-distance), MATLAB Central File Exchange. Retrieved January 15, 2021.

Dimitrievski, M., Van Hamme, D., Veelaert, P., & Philips, W. (2016). Robust matching of occupancy maps for odometry in autonomous vehicles. In *11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016)* (Vol. 3, pp. 626-633).

Do, M. N., & Vetterli, M. (2005). The contourlet transform: an efficient directional multiresolution image representation. *IEEE Transactions on image processing*, *14*(12), 2091-2106.

Dong, Z., Liang, F., Yang, B., Xu, Y., Zang, Y., Li, J., Wang, Y., Dai, W., Fan, H., Hyyppä, J. and Stilla, U. (2020). Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, *163*, 327-342.

Dryden, I. L., & Mardia, K. V. (1998). Statistical shape analysis: Wiley series in probability and statistics, Chichester: John Wiley & Sons, Inc.

Eiter, T., & Mannila, H. (1994). *Computing discrete Fréchet distance* . Technical Report CD-TR 94/64, Christian Doppler Laboratory for Expert Systems, TU Vienna, Austria, 636-637

Grant, D., Bethel, J., & Crawford, M. (2012). Point-to-plane registration of terrestrial laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, *72*, 16-26.

Grohs P., Keiper S., Kutyniok G., and Schafer M., \alpha -molecules, Appl. Comput. Harmon. Anal., 41 (2016), pp. 297--336, https://doi.org/10.1016/j.acha.2015.10.009.

Gruen, A. (2012). Development and status of image matching in photogrammetry. *The Photogrammetric Record*, *27*(137), 36-57.

Gruen, A., & Akca, D. (2005). Least squares 3D surface and curve matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, *59*(3), 151-174.

Gruen, A., & Stallmann, D. (1993). High accuracy edge matching with an extension of the MPGC-matching algorithm. In *Applications of Geodesy to Engineering* (pp. 339-350). Springer, Berlin, Heidelberg.

Guo, J., Wu, L., Zhang, M., Liu, S., & Sun, X. (2018). Towards automatic discontinuity trace extraction from rock mass point cloud without triangulation. *International Journal of Rock Mechanics and Mining Sciences*, *112*, 226-237.Gruszczyński, W., Matwij, W., & Ćwiąkała, P. (2017). Comparison of low-altitude UAV photogrammetry with terrestrial laser scanning as data-source methods for terrain covered in low vegetation. *ISPRS Journal of Photogrammetry and Remote Sensing*, *126*, 168-179.

Habib, A., Ghanma, M., Morgan, M., & Al-Ruzouq, R. (2005). Photogrammetric and LiDAR data registration using linear features. *Photogrammetric Engineering & Remote Sensing*, *71*(6), 699-707.

He F., & Habib, A. (2016). A closed-form solution for coarse registration of point clouds using linear features. *Journal of Surveying Engineering*, *142*(3), 04016006.

Holz, D., Ichim, A. E., Tombari, F., Rusu, R. B., & Behnke, S. (2015). Registration with the point cloud library: A modular framework for aligning in 3-D. *IEEE Robotics & Automation Magazine*, *22*(4), 110-124.

Jiang, X., Liu, M., Huang, Y., & Luo, R. (2020, November). Review on improved algorithms based on ICP algorithm. In *2020 International Conference on Computer Engineering and Intelligent Control (ICCEIC)* (pp. 185-189). IEEE.

Jolliffe, I. T (2002). *Principal Component Analysis*. 2nd ed., Springer, NY.

Kendall, D. G. (1989). A survey of the statistical theory of shape. *Statistical Science*, 4 (2), 87-99.

King, E. J., Reisenhofer, R., Kiefer, J., Lim, W. Q., Li, Z., & Heygster, G. (2015, September). Shearlet-based edge detection: flame fronts and tidal flats. In *Applications of Digital Image Processing XXXVIII* (Vol. 9599, p. 959905). International Society for Optics and Photonics.

Kovesi, P. (2019). MATLAB and Octave Functions for Computer Vision and Image Processing. Available from: <http://www.peterkovesi.com/matlabfns/>.

Kromer, R. A., Abellán, A., Hutchinson, D. J., Lato, M., Chanut, M. A., Dubois, L., & Jaboyedoff, M. (2017). Automated terrestrial laser scanning with near-real-time change detection–monitoring of the Séchilienne landslide. *Earth surface dynamics*, *5*(2), 293-310.

Kromer, R. A., Hutchinson, D. J., Lato, M. J., Gauthier, D., & Edwards, T. (2015). Identifying rock slope failure precursors using LiDAR for transportation corridor hazard management. *Engineering Geology*, *195*, 93-103.

Kromer, R., Walton, G., Gray, B., & Lato, M. (2019). Development and Optimization of an Automated Fixed-Location Time Lapse Photogrammetric Rock Slope Monitoring System. *Remote Sensing*, *11*(16), 1890.

Labate, D., Lim, W. Q., Kutyniok, G., & Weiss, G. (2005). Sparse multidimensional representation using shearlets. In *Optics & Photonics* (pp. 59140U-59140U). International Society for Optics and Photonics.

Lague, D., Brodu, N., & Leroux, J. (2013). Accurate 3D comparison of complex topography with terrestrial laser scanner: Application to the Rangitikei canyon (NZ). *ISPRS journal of photogrammetry and remote sensing*, *82*, 10-26.

Letortu, P., Jaud, M., Grandjean, P., Ammann, J., Costa, S., Maquaire, O., Davidson, R., Le Dantec, N. & Delacourt, C. (2018). Examining high-resolution survey methods for monitoring cliff erosion at an operational scale. *GIScience & Remote Sensing*, *55*(4), 457-476.

Li, J. (2003). *A wavelet approach to edge detection* (Master of Science, Sam Houston State University).

Li, J., Hu, Q., & Ai, M. (2020a). GESAC: Robust graph enhanced sample consensus for point cloud registration. *ISPRS Journal of Photogrammetry and Remote Sensing*, *167*, 363-374.

Li, J., Zhao, P., Hu, Q., & Ai, M. (2020b). Robust point cloud registration based on topological graph and Cauchy weighted lq-norm. *ISPRS Journal of Photogrammetry and Remote Sensing*, *160*, 244-259.

Li, X., Chen, Z., Chen, J., & Zhu, H. (2019). Automatic characterization of rock mass discontinuities using 3D point clouds. *Engineering Geology*.

Lowe, D. G. (1999, September). Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision* (Vol. 2, pp. 1150-1157). Ieee.

MathWorks, Inc (2020a). matchFeatures. https://www.mathworks.com/help/vision/ref/matchfeatures.html

MathWorks, Inc (2020b). pca. https://www.mathworks.com/help/stats/pca.html

Monserrat, O., & Crosetto, M. (2008). Deformation measurement using terrestrial laser scanning data and least squares 3D surface matching. *ISPRS Journal of Photogrammetry and Remote Sensing*, *63*(1), 142-154.

Novak, D., & Schindler, K. (2013). Approximate registration of point clouds with large scale differences. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, *1*(2), 211-216.

O'Banion, M. S., Olsen, M. J., Rault, C., Wartman, J., & Cunningham, K. (2018). Suitability of structure from motion for rock-slope assessment. *The Photogrammetric Record*, *33*(162), 217-242.

Pérez-Rey, I., Riquelme, A., González-deSantos, L. M., Estévez-Ventosa, X., Tomás, R., & Alejano, L. R. (2019). A multi-approach rockfall hazard assessment on a weathered granite natural rock slope. *Landslides*, 1-11.

Persad, R. A., & Armenakis, C. (2017). Automatic co-registration of 3D multi-sensor point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *130*, 162-186.

Pratt WK (1978) *Digital Image Processing*. John Wiley & Sons Inc. New York, USA.

Prewitt, J. M. (1970). Object enhancement and extraction. *Picture processing and Psychopictorics*, *10*(1), 15-19.

Reisenhofer, R., & King, E. J. (2019). Edge, Ridge, and Blob Detection with Symmetric Molecules. *SIAM Journal on Imaging Sciences*, *12*(4), 1585-1626.

Reisenhofer, R., Kiefer, J., & King, E. J. (2016). Shearlet-based detection of flame fronts. *Experiments in Fluids*, *57*(3), 41.

Richards, J. A., (2013). *Remote sensing digital image analysis*. Springer. 5th edition, Canberra, Australia.

Rouyet, L., Kristensen, L., Derron, M. H., Michoud, C., Blikra, L. H., Jaboyedoff, M., & Lauknes, T. R. (2017). Evidence of rock slope breathing using ground-based InSAR. *Geomorphology*, *289*, 152-169.

Rusu, R. B., Blodow, N., & Beetz, M. (2009, May). Fast point feature histograms (FPFH) for 3D registration. In *2009 IEEE International Conference on Robotics and Automation* (pp. 3212-3217). IEEE.

Shen, L., & Bai, L. (2006). A review on Gabor wavelets for face recognition. *Pattern analysis and applications*, *9*(2-3), 273-292.

Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (4), 376-380.

Van Veen, M., Hutchinson, D. J., Kromer, R., Lato, M., & Edwards, T. (2017). Effects of sampling interval on the frequency-magnitude relationship of rockfalls detected from terrestrial laser scanning using semi-automated methods. *Landslides*, *14*(5), 1579-1592.

Von Hansen, W. (2006). Robust automatic marker-free registration of terrestrial scan data. *Proc. Photogramm. Comput. Vis*, *36*, 105-110.

Williams, J. G., Rosser, N. J., Hardy, R. J., Brain, M. J., & Afana, A. A. (2018). Optimising 4-D surface change detection: an approach for capturing rockfall magnitude–frequency. *Earth surface dynamics.*, *6*(1), 101-119.

Xian, Y., Xiao, J., & Wang, Y. (2019). A fast registration algorithm of rock point cloud based on spherical projection and feature extraction. *Frontiers of Computer Science*, *13*(1), 170-182.

Yin, C., Li, H., Hu, Z., & Li, Y. (2020). Application of the Terrestrial Laser Scanning in Slope Deformation Monitoring: Taking a Highway Slope as an Example. *Applied Sciences*, *10*(8), 2808.

Zieher, T., Toschi, I., Remondino, F., Rutzinger, M., Kofler, C., Mejia-Aguillar, A., & Schlögel, R. (2018). Sensor-and scene-guided integration of TLS and photogrammetric point clouds for landslide monitoring. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-2, 2018 ISPRS TC II Mid-term Symposium “Towards Photogrammetry 2020”, 4–7 June 2018, Riva del Garda, Italy

Zhang, J., Tang, B., & Yiu, M. L. (2019). Fast Trajectory Range Query with Discrete Frechet Distance. In *EDBT* (pp. 634-637).

Zhang, J., Yao, Y., & Deng, B. (2021). Fast and Robust Iterative Closest Point. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.