Journal of Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

Maximizing feature detection in aerial unmanned aerial vehicle datasets

Jonathan Byrne
Debra F. Laefer
Evan O'Keeffe

SPIE.

Jonathan Byrne, Debra F. Laefer, Evan O'Keeffe, "Maximizing feature detection in aerial unmanned aerial vehicle datasets," *J. Appl. Remote Sens.* **11**(2), 025015 (2017), doi: 10.1117/1.JRS.11.025015.

Maximizing feature detection in aerial unmanned aerial vehicle datasets

Jonathan Byrne,[a] Debra F. Laefer,[a,b,*] and Evan O'Keeffe[a,c]
[a]University College Dublin, School of Civil Engineering, Urban Modelling Group, Belfield, Dublin, Ireland
[b]New York University, Center for Urban Science and Progress, Brooklyn, United States
[c]University College Dublin, School of Computer Science and Informatics, Belfield, Dublin, Ireland

Abstract. This paper compares several feature detectors applied to imagery from an unmanned aerial vehicle to find the best detection algorithm when applied to datasets that vary in translation and have little or no image overlap. Metrics of inliers and reconstruction accuracy of feature detectors are considered with respect to three-dimensional reconstruction results. The image matching results are tested experimentally, and an approach to detecting false matches is outlined. Results showed that although the detectors varied in the number of keypoints generated, a large number of inliers does not necessarily translate into more points in the final point cloud reconstruction and that the process of comparing a large quantity of redundant keypoints may outweigh the advantage of having the extra points. The results also showed that despite the development of keypoint detectors and descriptors, none of them consistently demonstrated a substantial improvement in the quality of structure from motion reconstruction when applied to a wide range of disparate urban and rural images. [DOI: 10.1117/1.JRS.11.025015]

Keywords: remote sensing; image segmentation; photogrammetry; detection; computer vision.

Paper 16969 received Dec. 19, 2016; accepted for publication Jun. 6, 2017; published online Jun. 27, 2017.

1 Introduction

Aerial data are now nearly ubiquitous due to the popularity of low-cost, unmanned aerial vehicles (UAVs).[1,2] While there are similarities between aircraft and UAV surveys, they differ in some important aspects that complicate the use of traditional photogrammetry-based solutions for three-dimensional (3-D) reconstructions.

First, traditional aerial surveys are normally flown at heights of 500 to 1500 m with the exact position recorded by an inertial navigation system. In contrast, UAV surveys are flown at heights of 30 to 120 m and have, at best, a recorded global positioning system (GPS) coordinate with an accuracy of 5 to 50 m.

Secondly, aerial survey images are normally collected within a tightly controlled flightpath with the camera pointing nadir and with a 50% to 80% overlap of the target (Fig. 1). The images only vary by translation, with no changes to the scale and rotation. In contrast, UAV cameras may have a variety of orientations with respect to the surrounding environment. Although the amount of overlap between images can be controlled by autopiloting software, when captured manually by the pilot, the overlap can be haphazard and vary greatly.

A further challenge when reconstructing UAV surveys is the height at which the data are captured. The UAVs' lower altitudes cause much smaller areas to be covered, which reduces the number of distinct features available for detection by a matching algorithm. In highly convergent imagery, the same object can appear in most or all of the images. An object in nadir

*Address all correspondence to: Debra F. Laefer, E-mail: debra.laefer@nyu.edu

1931-3195/2017/$25.00

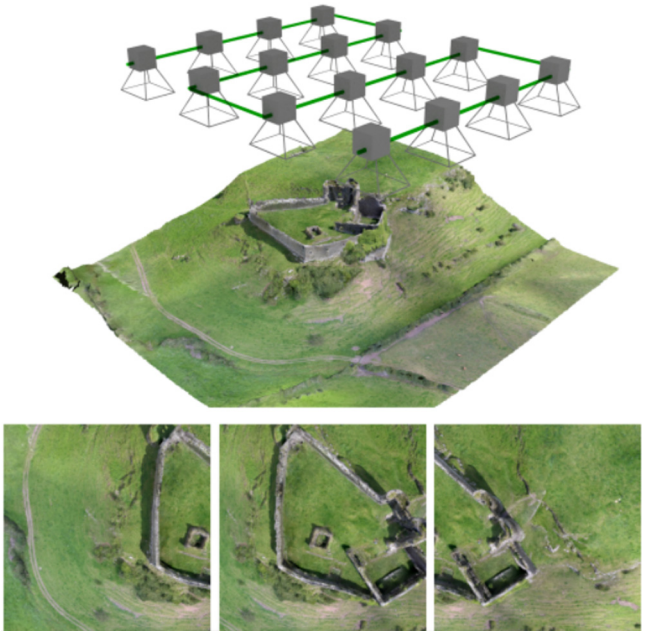


Fig. 1 Traditional aerial photography captures images with the camera pointing vertically downward with the same scale and rotation in each image.

oriented images from a UAV may have significant overlap with neighboring images but could be in only a small subset of the total image set.

Additionally, when an image is captured from 1000 m above, there is little parallax, as the variation in height (0 to 100 m) is small. Instead, if an image is captured from only 100 m above the ground, the relative distance of the camera has a major impact. As an example, the tall structure in Fig. 2 looms in the foreground and distorts significantly over successive images.

The perspective of the camera lens also further distorts the image. Standard orthorectification approaches such as photostitching[3] fail at low altitudes[4] due to the reasons given above. These distortions make mapping UAV imagery much more challenging than traditional aerial data sets.

Modern aerial mapping approaches have borrowed from computer vision and robotics to combine images, generate orthomosaics, and reconstruct full 3-D models. One of the most successful techniques is structure from motion (SFM).[5] SFM creates 3-D models purely from images, with no reliance on *a priori* information or locational data. Feature detection is a critical step in the reconstruction of 3-D models from such imagery. Feature detection algorithms generally consist of two parts. The first is a detector that finds features in an image, such as corners and edges. These can be used to find corresponding matches in other images. The second relates to matching and involves a descriptor that condenses information about a point into a format that enables accurate identification or, at a minimum, a measure of similarity between points.

One main difference between UAV datasets and computer vision datasets is the amount of matching that is required. Normally, SFM is applied to small sets of images captured sequentially and assumes that there is a local overlap. UAV datasets are much larger, and a feature may be

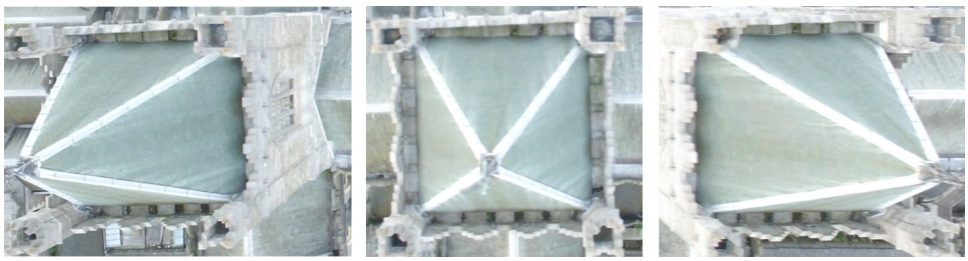


Fig. 2 The parallax of a church spire captured at a low altitude.

present in a small subset of the images. Notably, global comparisons over the dataset can result in false matches, which invalidate the reconstruction. The false positive rate of the feature descriptor (the sequence of bytes defining the feature) is examined in this work, as it is a significant problem with UAV datasets.

The intention of this work is to compare different feature detectors on UAV datasets to establish which generate the greatest quantity of features with the highest accuracy. Although the speed of a feature detector is always important, this is less of a constraint with 3-D model reconstruction, as it is usually conducted offline, after the flight. In contrast, a critical aspect is the number of feature matches—the component that generates an accurate 3-D model. Herein, the feature set generated for an image is not considered in isolation. Instead, it is compared against a feature set from another image, and the number of matches is computed.

While matches may be an indication of overlap, this is entirely dependent on the content and character of the area in the image. Accordingly, additional metrics are used to compute matching quality. This paper analyzes the metrics of inliers and reconstruction accuracy of feature detectors when applied to vertical image-based datasets captured at low altitudes to examine which technique provides the best results for 3-D reconstruction from UAV datasets.

2 Background

SFM is used extensively in aerial surveying to generate 3-D models. The technique consists of four components: feature detection, feature matching, bundle adjustment, and multiview stereo. A feature is essentially an "interesting" or easily identifiable part of an image. As mentioned above, a feature could be an edge, corner, or region of interest. Feature detectors abstract information from within an image into a subset of objects that can then be identified in multiple, overlapping images. Detectors are used to alleviate the complexity of processing an entire image while exploiting local appearance properties. In contrast, the ideal feature descriptor captures the most important and distinctive information content enclosed in the detected salient regions, such that the same structure can be recognized if re-encountered.

Originally developed for simultaneous localization and mapping for robotics, SFM is normally used with highly localized data sets. Images are typically only compared with others taken in close proximity to each other. As such, arguably some level of overlap has traditionally been assumed. In contrast, aerial datasets require global matching comparisons over the entirety of the image set. This presents a challenge, as many of the images have no overlap whatsoever. Consequently, a robust feature descriptor is required so that images with no overlap are not matched accidentally.

2.1 *Feature Detectors*

This section gives an overview of the feature detectors that are used in this work. The implementations used are taken from the OpenCV library,[6] and the matching and 3-D reconstructions employ the OpenMVG framework.[7] The five detectors presented herein are a combination of some of the most popular and some of the most recent additions in computer vision. Both proprietary and open source feature detectors are compared including scale invariant feature transforms (SIFT), speeded up robust features (SURF), oriented features from accelerated segment test (FAST), rotated binary robust-independent elementary features (BRIEF), and accelerated KAZE (AKAZE).

2.1.1 *Scale invariant feature transforms*

SIFT builds a feature description from the image that is invariant to orientation and uniform scaling and is robust to partial occlusion and lighting changes.[8] SIFT itself uses several different algorithms to build a set of feature vectors (Fig. 3). First, SIFT creates a scale space of images using a difference of Gaussians algorithm.[9] The algorithm compares blurred pixels at different image scales. These features are then evaluated using a contrast threshold and discards points below a prespecified threshold. A histogram of gradients (HoG) is generated for each feature, which is then used as a feature vector descriptor. A database of unique features is built for each

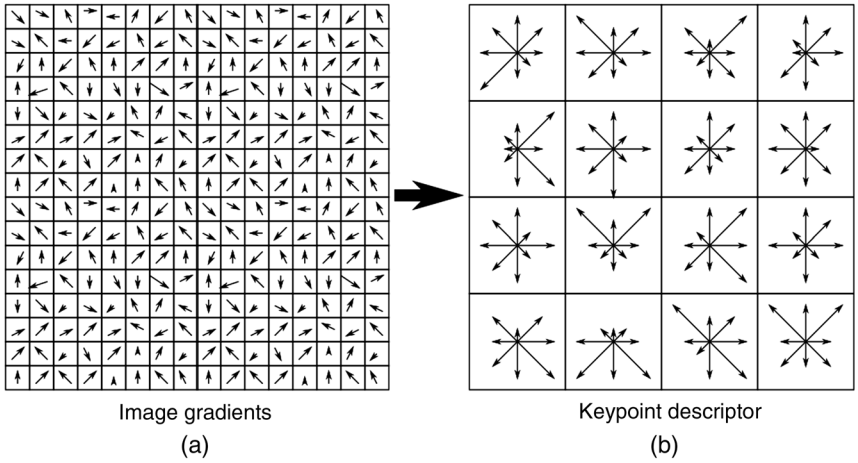


Fig. 3 Image gradients are divided into 16^2, and the orientations are then summed. Each square is represented by an 8-byte descriptor.

image, which can then be used to find matching points between images. While this is one of the oldest methods, it is still one of the most effective.

However, there are disadvantages with SIFT. The first relates to Gaussian blurring to find features (i.e., the use of an averaging kernel function to blur an image). This approach at coarser scales removes features with noise and finds more prominent structures but does not respect the boundary between actual objects and noise. The second is use of a HoG descriptor that is 128 bytes long, which makes it computationally less efficient than other descriptors. If time is not a consideration, then this approach can be very attractive due to its generally high level of output accuracy. The third drawback is that the algorithm is patent protected and requires licensing for commercial usage.

2.1.2 *Speeded up robust features*

SURF was the first algorithm to have a comparable matching performance to SIFT with the advantage of improved speed.[10] The acceleration was achieved by using the image integral, instead of a Gaussian pyramid, to find points of interest. The Gaussian pyramid requires different scale levels to be calculated for the image. This is approximated in SURF by using a box filter on the image integral to generate the scales (Fig. 4). The descriptor is similar to SIFT in that it uses a HoG that is calculated using Haar wavelet responses. While faster than SIFT, SURF still suffers from many of SIFT's problems in that the scale space loses boundary information and that the descriptors are long (normally 64 or 128 bytes). SURF is also patent protected.

2.1.3 *Oriented FAST and rotated BRIEF*

The oriented FAST and rotated BRIEF (ORB) method was developed as a simple and fast alternative to SIFT and SURF.[11] Instead of costly, scale-space calculations and HoG descriptors, ORB uses a binary-based, feature detector, and descriptor. The detector is a generalization

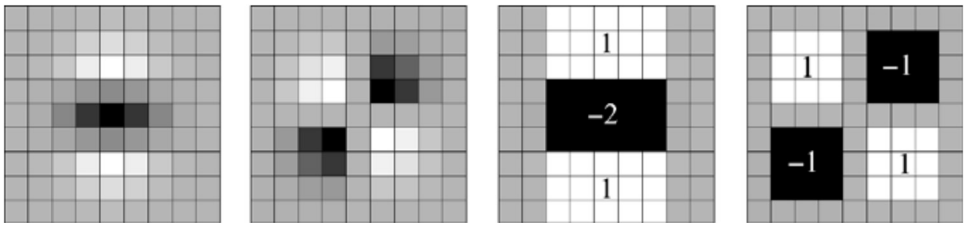


Fig. 4 Gaussian second-order partial derivatives in (a) the y-direction and the xy-direction, and (b) the corresponding box filter approximation.[10]

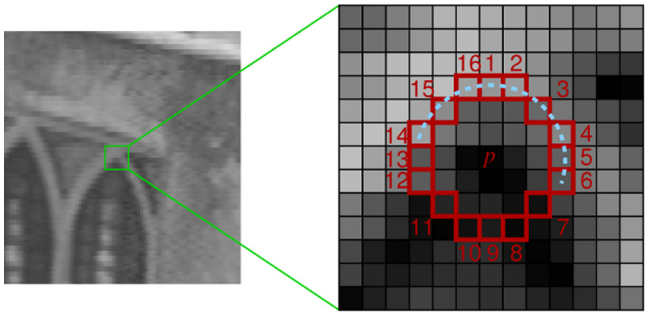


Fig. 5 The fast corner detector (image courtesy of OpenCV[6]).

on the features FAST algorithm[12] combined with a descriptor based on BRIEF.[13] The idea behind ORB was to develop a fast and open source alternative to existing feature detectors.

The FAST algorithm was developed for real-time odometry from video footage; a task that was not possible with more computationally expensive detectors like the Harris corner detector[14] or SIFT. Instead, ORB finds corners and edges through performing a simple test by considering a circle of sixteen pixels around a corner candidate. If a set of n contiguous pixels in a circle exist that are all brighter than the intensity of the candidate pixel or all darker, then n is chosen to be 12, because it demonstrates that a high-speed test can be used to exclude a very large number of noncorners. The test examines only the four pixels at locations 1, 5, 9, and 13 (the four compass directions) (Fig. 5).

If p is a corner and Ip is the pixel intensity, then at least three of these must be brighter than $Ip + t$ or darker than $Ip - t$. If neither is the case, then p cannot be a corner. The corner detector was improved in ORB by computing it at different image scales for scale invariance, and a Harris edge detector was also applied to remove noisy edges. One component that is missing from FAST is that it does not compute the orientation of the point of interest. SIFT does this by using the HoG, which is expensive. SURF approximates it using block patterns, which yields poor approximations. Instead, ORB adds an orientation by using a centroid approximation as detailed by Rosin.[15] Some improvements were made to the descriptor as well. For example, the original BRIEF algorithm was not designed to be rotation invariant, and so Rublee et al.[11] incorporated orientation information derived from the keypoints generated by FAST and integrated it as a rotational element into the descriptor, now known as rBRIEF.

2.1.4 *Accelerated KAZE*

AKAZE improves the idea of using a scale space for identifying features at distinct scales[16] by using a nonlinear diffusion filter. The difference of the Gaussian pyramid used by SIFT to create a scale space of images is one of the simplest approaches but performs poorly in localization accuracy. This is because while the coarser (more blurred) scales remove noise and emphasize more prominent features, it complicates identifying the actual location (Fig. 6). AKAZE, instead,

Fig. 6 Comparison between (a) Gaussian and (b) nonlinear diffusion filtering. Note the increased blurring in the Gaussian filtering (image courtesy of Ref. 16).

uses a nonlinear diffusion filter called adaptive operator splitting that creates locally adaptive blurring. This nonlinear filter removes the noise while maintaining details and edges. Nonlinear filtering requires solutions to the partial differential equations (PDE) that define them. As no analytical solution to PDEs exists, a numerical method such as the Thomas algorithm[17] is used to iteratively generate a solution. Once the scale pyramid has been created, the first- and second-order derivatives are approximated using a 3×3 Scharr filter,[18] which approximates rotation invariance. The descriptor first finds the orientation in a similar fashion to SURF by finding the dominant orientation in a circular area or radius of the sample step size. The descriptor format is the modified SURF descriptor modified for a nonlinear scale space.

2.1.5 *Binary robust invariant scalable keypoints*

Binary robust invariant scalable keypoints (BRISK) claims to be a feature detector that produces high-quality performance at a much lower computational cost.[19] The method uses a variation on the FAST detector that is adapted to use a scale space after finding a suitable key point. The descriptor consists of a bit-string that records intensity comparisons for the feature neighborhood and is rotation and scale invariant (Fig. 7). The detection methodology applies the adaptive and generic accelerated segment test,[20] an accelerated version of FAST, to both the image plane and the scale space. The fast score $\mathbf{s}$ is used as a measure of saliency (i.e., how well it stands out relative to its neighboring pixels). The scale space pyramid consists of n octaves and n intra-octaves. The octaves are formed by half sampling the original image, and the intraoctaves are obtained by down-sampling the original image by 1.5. A FAST 9×16 detector is used on each layer, which requires 9 consecutive pixels in the 16-pixel circle to be sufficiently brighter or darker than the central pixel for the FAST criterion to be evaluated. The correct octave on the scale space pyramid for the key point is determined by comparing the saliency scores in the immediate neighboring layers.

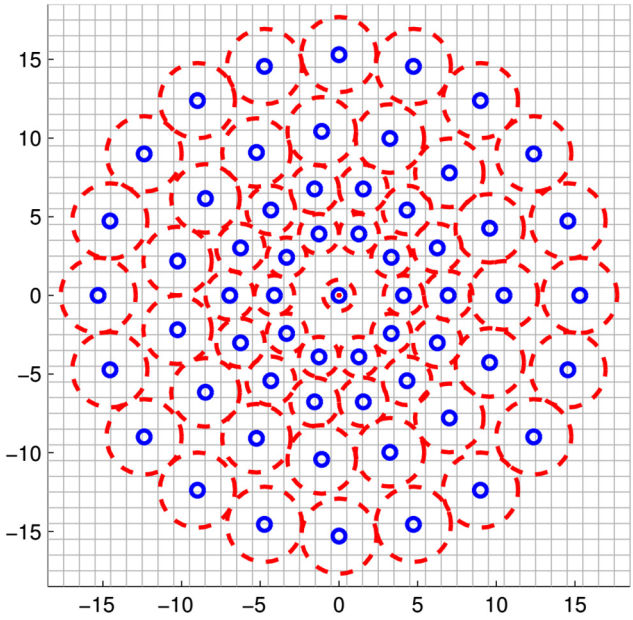


Fig. 7 BRISK sampling pattern with $N = 60$ points. Blue circles are the sampling locations, and red circles correspond to the standard deviation of the Gaussian kernel used to smooth the intensity of the sampling points (image courtesy of Ref. 19). The feature descriptor is modeled on the BRIEF binary descriptor[21] that concatenates the results of brightness comparison tests. To maintain rotation invariance, the characteristic direction of each key point is calculated. This allows for an orientation normalized descriptor.

Fig. 8 Richview buildings.

3 Datasets

The intention of this work was to examine how well each detector matches images that vary through translation. To achieve this objective, three controlled datasets from Dublin Ireland were considered. The areas included the Richview portion of the University College Dublin campus, a glasshouse on the Kilmoon Cross Farm on the outskirts of Dublin, and Boland's Mills, a 19th century industrial building in Dublin's city center. The Richview buildings provided a standard site survey with a series of low-level buildings. The Kilmoon Cross farm site consisted of a set of glasshouses; although there was little parallax, the reflections added noise to the feature detection task. Finally, Boland's Mills was a site survey of a set of tall buildings that introduced significant levels of parallax. The datasets are discussed in more detail below.

3.1 *Richview Buildings*

The Richview building complex is home to the University College Dublin's School of Architecture, Planning, and Environmental Policy (Fig. 8). This portion of the campus has been extensively surveyed by total station and by terrestrial laser scanning, which allows verification of the accuracy of the resulting orthomosaic. The Richview site was chosen, because it has buildings with numerous, varied features, such as corners and edges of complicated building profiles. The building heights range between 6 and 30 m. The images were taken from 30, 40, and 50 m above ground level. There were few reflective surfaces to introduce noise into the algorithm.

3.2 *Kilmoon Cross Farm*

The Kilmoon Cross Farm glasshouses provided a particularly difficult challenge for orthorectification, as the structures were uniform, and the glass was highly reflective (Fig. 9). The reflections generated transient keypoints that changed from image to image. Uniformity in a structure resulted in improper matching, as different homography matrices produced seemingly good results through incorrect alignment. The data were captured at a height of 70 m above the ground.

3.3 *Boland's Mills*

Boland's Mills is a site of historic significance in Dublin's city center. The first structure was built in 1830, with further concrete silos built between the 1940s and the 1960s. The mill ceased operation in 2001 and is now undergoing a 150-million-euro conversion into office spaces and residential housing. As several of the buildings in the complex are listed in various historic registries, an aerial survey was conducted to create a permanent digital record of the site.

Fig. 9 Kilmoon Cross glasshouses.

Fig. 10 Boland's Mills.

Boland's Mills has several relatively tall structures (55 m) that introduced a large amount of parallax between images (Fig. 10). The survey was conducted at a height of 85 m, which was 30 m above the rooftop of the highest building. Data could only be captured from a single height due to the congestion of the built environment and the proximity to the Dublin airport. These factors controlled both the minimum and maximum possible flight altitudes.

4 Experimental Setup

The images were obtained using a Phantom 3 Professional quadcopter, a commercial off-the-shelf UAV. A Sony Exmor, fixed aperture, 12-megapixel camera was used in this work. The resulting images were 4000×3000 pixels. At a height of 50 m, this translated into a ground sampling distance of 2 cm/pixel. The camera had a 94-deg field of view and a focal length of 3.61 mm. It had a self-stabilizing 3-axis gimbal that ensured the nadir images were correctly oriented.[22] The Sony Exmor camera used in these experiments provided 12-megapixel images with only a small amount of rectilinear distortion.[22] The Phantom 3 recorded metadata onto the images including GPS coordinates. SFM uses the camera settings saved in the metadata, such as the focal length and camera model as input in the reconstruction process. SFM does not use the GPS coordinates in the reconstruction process, although they may be used afterward to geo-locate the model. The same camera was used for gathering all of the datasets.[23]

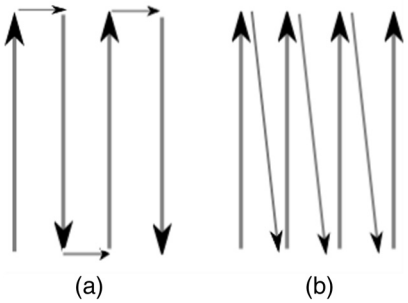


Fig. 11 Flight paths: (a) serpentine and (b) zig-zag.

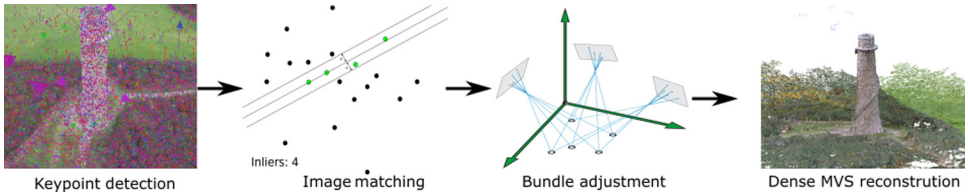


Fig. 12 SFM pipeline for generating a point cloud.

In order to provide consistency between datasets, each was acquired at a specific height with an 80% vertical and 60% horizontal overlap. This was accomplished using the Pix4D capture mapper software[24] that automatically generated the flightpath and images for a given survey area. The images were acquired vertically downward, with the gimbal facing toward earth to ensure there was no variance due to turbulence. A serpentine (back and forth) path was flown [Fig. 11(a)] for the missions, rather than a zig-zag flight path [Fig. 11(b)]. All flights per site were conducted on the same day and within minutes of each other so as to control for lighting and cloud-based occlusion, which are known to generate problems in image processing.[24]

The SFM pipeline consists of feature detection, matching, bundle adjustment, and dense reconstruction (Fig. 12). Once the keypoints were detected, a process such as random sampling and consensus (RANSAC)[25] was used to compute the best match between images. In this work, every image in the dataset was compared with each other to ensure the best possible match. Once this was complete, a 3-D scene was constructed by computing the bundle adjustment[26] for all the cameras and matched points. Finally, once the 3-D scene was created, a final step of dense reconstruction through multiview stereo[27] was used to generate a dense point cloud. The accuracy was measured in two ways: (1) inlier matching and (2) final reconstruction.

5 Evaluation

5.1 *Inlier Matching*

The set of features defined by SIFT can contain outliers or points not common to both images (depending on the overlap). However, the computational expense of comparing every feature in an image increases exponentially, as the number of points increases. Additionally, noise from outliers cannot be handled with simplistic data fitting models such as the method of least squares, which minimizes the sum of squared distances between an observed point and the fitted value provided by the model.

Instead, RANSAC, a robust iterative technique that constructs an alignment model in linear time, is used. RANSAC assumes that the data contain "inliers" that fit within a given model and "outliers" (noise) that do not fit within that model. The algorithm samples enough points to minimally fit the model and measures the number of inliers and outliers within a given threshold, as shown in Fig. 13. The process is repeated a set number of times to build a model. The approach is probabilistic. Thus, perfect alignment is not ensured, but it is robust against noise, although the computation time increases linearly when aligning images.

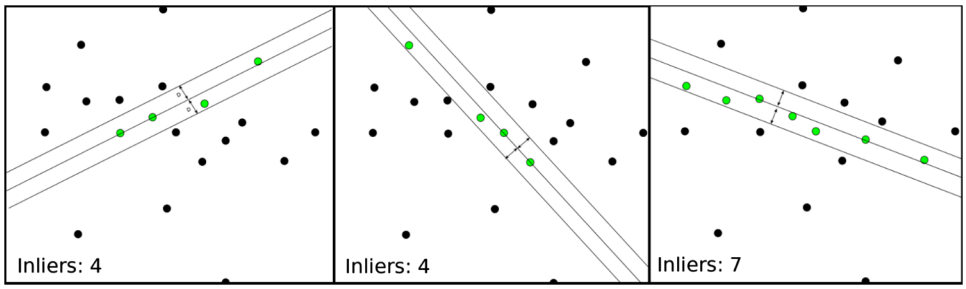


Fig. 13 An example of line fitting to a noisy dataset using RANSAC. Two points are chosen to create the line, and then the number of inliers within a given threshold are counted.

RANSAC generates an inlier count for each image comparison. The example given in Fig. 13 is fitting a line to a set of points, whereas the model required for SFM matching requires RANSAC to construct an eight-point alignment model in linear time. The greater number of inliers is indicative of a more accurate match. This measure is normally used when comparing how well a feature detector performs.[28]

5.2 *Bundle Adjustment and Dense Reconstruction*

The next step is bundle adjustment. Bundle adjustment refines a visual reconstruction to jointly produce the optimal 3-D structure and the viewing parameters. The "bundle" refers to the bundle of light rays leaving each 3-D feature and converging on each camera center, as shown in Fig. 14. The bundles are optimally adjusted with respect to both feature and camera positions. Bundle adjustment must occur after outlier removal, as the process is sensitive to noise.

Bundle adjustment treats the reconstruction as an optimization problem that aims to minimize the reprojection error between noisy images. The viewing parameters and 3-D reconstruction parameters are considered equivalent and solved simultaneously. The adjustment assumes that the data are distributed normally and only contain small errors. The distribution of many small random deviations almost always converges to a normal distribution, as stated in the central limit theorem. The problem is then treated as a least squares problem, and the squared error for the bundles is reduced iteratively. An example of the result after bundle adjustment is shown in Fig. 15. The bundle adjustment output is a sparse point cloud, and the flight path is visible as the image frustums have been aligned by the process.

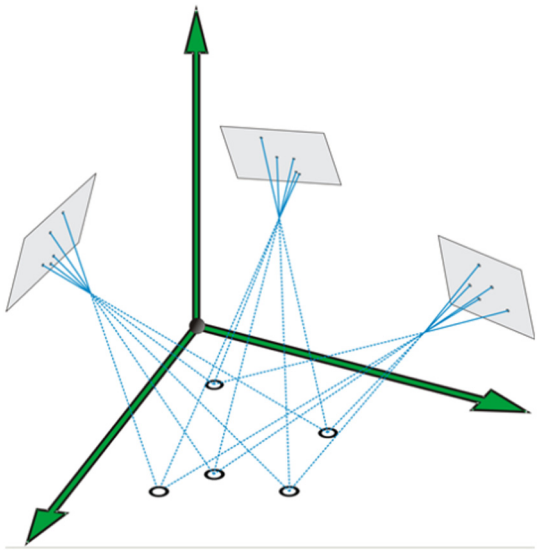


Fig. 14 A bundle of light rays is projected from the frustum to the points in the scene and the error is minimized.

Fig. 15 Sparse reconstruction after bundle adjustment versus the dense reconstruction after MVS.

The sparse point cloud model generated from SFM can be thickened by interpolation. The technique called multiview stereopsis (MVS) uses points that have already been matched and interpolates points using image data. Once additional points have been generated, visibility filters are applied to remove invalid points. MVS first decomposes the input images into a set of image clusters of manageable size so that dense reconstruction can occur. The algorithm then matches them across multiple images to form an initial set of patches and uses an expansion procedure to obtain a denser set of patches, before assigning visibility constraints (e.g., visibility consistency, depth map test) to filter any false matches. The process generates a much denser cloud but is dependent on the quality and density of the initial keypoint set. Using MVS generates a much richer point set than the sparse reconstruction created by the bundle adjustment, as demonstrated in Fig. 15.

5.3 *False Positive Detection*

Normally, feature comparisons are done on small datasets (5 to 20 images) where there is significant overlap between all the images, and the homography matrices are known. This is not the case with low-altitude UAV datasets that cover a large area, as the features may only be in a small subset of the total set of images. As such, a feature detector that finds a large amount of "inliers" where there should be none is detrimental to the image matching process. As such, this paper introduces a measure for comparing feature detectors where there is little or no image overlap. This is achieved by finding the quantity of false matches and false inliers. To accomplish this, the GPS information recorded in the exchangeable image file metadata was used to calculate overlap between two images.

The coordinates for each image are recorded in World Geodetic System WGS84. This is an unprojected coordinate system, which means the coordinates are not projected onto a flat surface but instead, reference positions on the surface of a spheroid. The distance is effected by the both the curvature of the earth and the longitude of the location (1 deg of longitude is 111 km at the equator and shrinks to zero at the poles), which makes it difficult to measure distance. The Haversine equation[29] was used to approximate the distance between the images, and the FOV and height of the UAV were used to calculate the distance to the edge of the image. The maximum distance covered by the images and the distance between the image centers were used to calculate the overlap between images. These were then matched separately to evaluate whether the feature detector generates a significant number of false positives.

5.4 *OpenCV and OpenMVG*

All implementations for the feature detectors were taken from the OpenCV library[6] and used with the default settings except ORB, which in OpenCV has an arbitrary limit of 500 keypoints per image. ORB uses FAST, which uses a threshold for evaluating keypoints. A low threshold generates a large amount of keypoints that would normally slow down keypoint matching too much for real-time applications. So as to ensure a fair comparison, this limitation was removed. The feature detectors were applied to each image in each dataset to generate a set of keypoints. The keypoints were then globally matched against every other set of keypoints in the dataset to ensure that the best match correspondence was found. The keypoints were then compared against

the set of images where there was no overlap, in order to analyze how many false positive matches were generated.

The second test used the matched points to generate a 3-D reconstruction. Although most feature descriptor comparisons focus on keypoints generated and number of inliers, the example shown here demonstrates that these do not necessarily translate into an accurate model. The final output from the bundle adjustment step was a sparse point cloud. The resulting number of points provided a metric for evaluating how many keypoints comprised the final model. The OpenMVG framework[30] was used to generate the 3-D models. It is a powerful open source framework that allows for easy modification. The framework was adapted to use the OpenCV feature detectors to ensure an equitable comparison. The default matching settings were used to compute the matches, and the global scene reconstruction approach[7] was used to generate the sparse point cloud.

5.4.1 *Feature matching results*

The number of keypoints and matches were recorded for each image in the dataset. The results are shown in the boxplots in Figs. 16–22. The boxplots display the average times, the standard deviations, and outliers for the images. The number of matches and the inliers were generated by

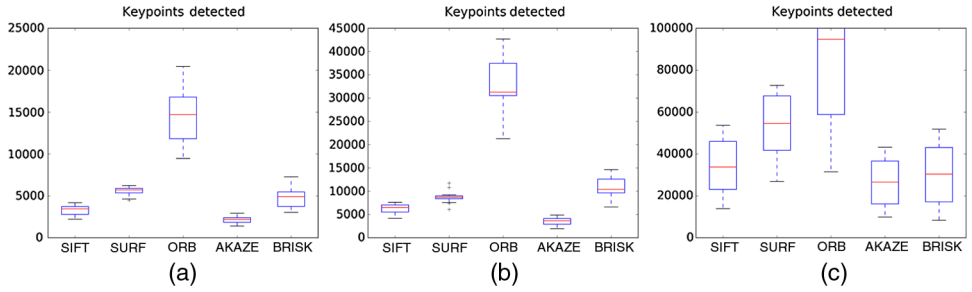


Fig. 16 Key points for nadir dataset: (a) Richview, (b) Kilmoon Cross, and (c) Boland's Mills.

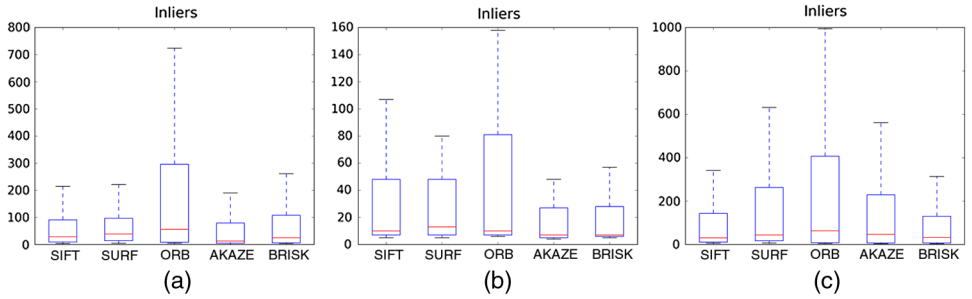


Fig. 17 Inlier results for nadir dataset: (a) Richview, (b) Kilmoon Cross, and (c) Boland's Mills.

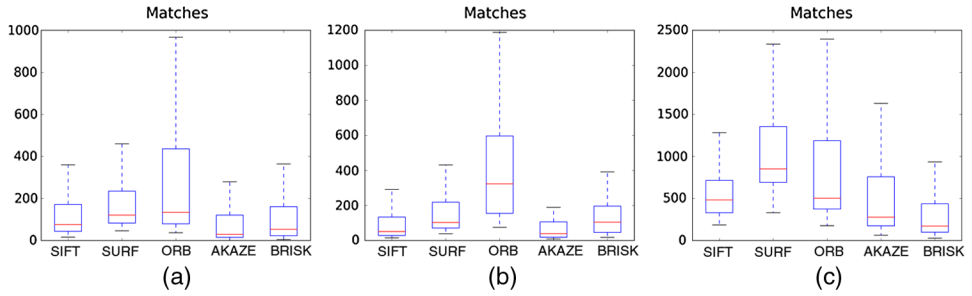


Fig. 18 Ratio matching results for nadir dataset: (a) Richview, (b) Kilmoon Cross, and (c) Boland's Mills.

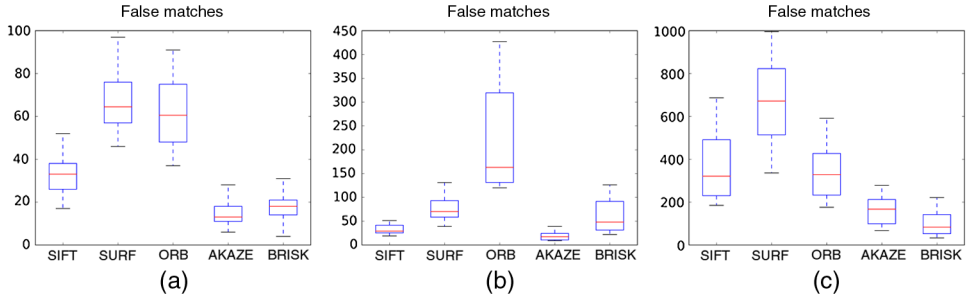


Fig. 19 False positive matching results for nadir dataset: (a) Richview, (b) Kilmoon Cross, and (c) Boland's Mills.

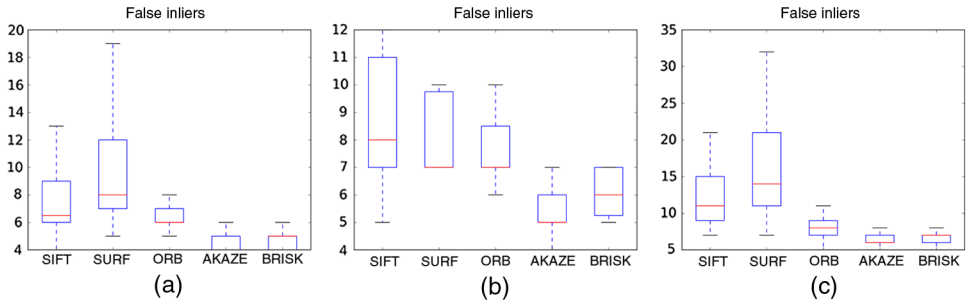


Fig. 20 False inlier results for nadir dataset: (a) Richview, (b) Kilmoon Cross, and (c) Boland's Mills.

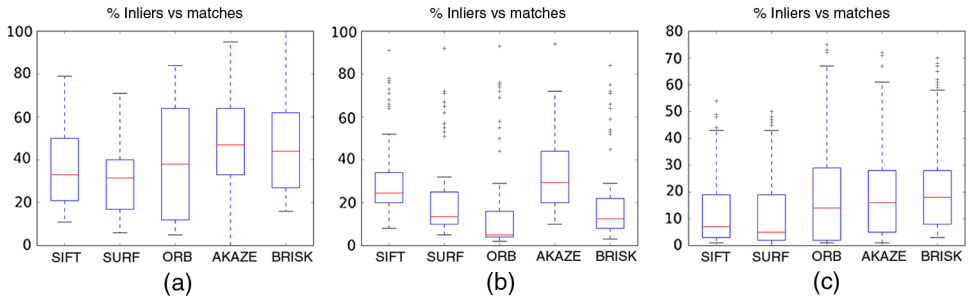


Fig. 21 Ratio of inliers results for nadir dataset: (a) Richview, (b) Kilmoon Cross, and (c) Boland's Mills.

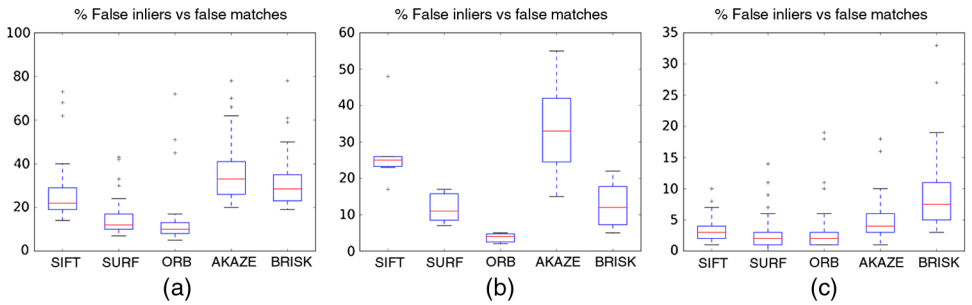


Fig. 22 Ratio of false inliers to false matches for nadir dataset: (a) Richview, (b) Kilmoon Cross, and (c) Boland's Mills.

comparing the keypoints of every image with every other image in the dataset. The same was then done for the false positive subsets for each image to analyze the global matching ability.

Each detector was compared with several metrics: (1) detected keypoints, (2) matches, (3) inliers, and (4) ratio of inliers to matches. The number of keypoints is self-explanatory. The number of matches was calculated by brute force matching of the keypoints in one set with all the keypoints in the second set. In the case of the integer descriptors of SIFT and SURF, this was calculated using a Euclidean distance equation. The distance for binary descriptors was computed using a Hamming distance. The closest match for each key point was found. The matches were filtered using the ratio test described as by Lowe;[8] if the second closest match was closer than the given ratio, then the match was considered valid. In these tests, the ratio was set to 0.75; the same as used by Lowe.[8]

The ratio test is only heuristic for calculating a match that contains many outliers and points that are not common to both images. A more robust technique is required to find an accurate match and calculate the overlap. The computational expense to compare every feature in an image increases exponentially, as the number of points increases. Instead, herein, points were matched by using the iterative technique RANSAC,[25] which constructs an alignment mode in linear time. RANSAC assumes that the data contain "inliers" that fit within a given threshold and "outliers" (noise) that do not fit within that model. The algorithm samples a sufficient number of points to fit the model minimally and measures the number of inliers and outliers. The approach is probabilistic; thus, perfect alignment is not ensured. RANSAC is used as it is robust against noise, and the computation time increases linearly when aligning images. OpenMVG uses a Contrario RANSAC,[31] which adaptively adjusts the threshold based on the amount of noise in the dataset.

5.4.2 *Inlier results*

The results shown in this section used RANSAC to match the images and count the number of inliers between them. The vertical axis is the number of inliers that were found when matching the keypoints. The number of false inliers was also checked by (1) comparing images where there was no overlap and (2) recording the number of detected matches.

ORB found the greatest number of keypoints in all the nadir datasets by a wide margin, followed by SURF or BRISK and then SIFT. AKAZE produced the least amount of keypoints. The Boland's Mills dataset proved an interesting dataset, generating up to 100,000 keypoints per image. This was due both to the numerous buildings and vehicles on site, as well as having images captured from the greatest height, with respect to the ground level. AKAZE generated a comparable number of keypoints to SIFT and BRISK on the Boland's Mills dataset but underperformed on the other datasets.

The results in Fig. 17 show the total number of inliers found by each of the detectors. These results are the most important with regards to generating a high-quality model, as the inliers are used as input for the bundle adjustment and dense reconstruction. Although the total number of inliers does not necessarily translate into an accuracy prediction for the reconstruction, it gives a measure of how many points will be included in the final reconstruction. For Richview, ORB generated the most inliers and AKAZE generated the least. Typically, the methods SIFT, SURF, and BRISK performed equally well. For Kilmoon Cross, the number of matches decreased significantly with each detector on average producing only 15 to 20 matches as compared to as many as 50 for the other data sets. For Boland's Mills, ORB generated the most inliers and SIFT the least.

Although the ratio test for rough matching is essentially heuristic, it removes many of the spurious results and gives a more accurate matching figure. The results (as shown in Fig. 18) imply that the majority of keypoints generated by ORB were not used. The inlier results in Fig. 17 fall into a similar pattern as the ratio matching results. Namely, the likelihood that a match will be found increases with more keypoints. In this, SURF performed best on the Boland's Mills dataset.

The false positive results shown in Fig. 19 demonstrate that ORB was more susceptible to noise such as reflections, whereas SURF had more false matches for the Boland's Mills dataset. AKAZE performed better on the noisy glasshouse data (Kilmoon Cross) but worse on the

Boland's Mills data. BRISK performed similarly on all datasets with a small number of false inliers. SIFT had very few false matches on the noisy glasshouse data but a large amount on the Boland's Mills dataset. This high level of inconsistency shows the large impact of the datasets themselves.

Figure 20 shows the number of false inliers, where there was no overlap between matched images. For the Richview case, SURF performed the worst, despite not initially generating a large number of inliers. In contrast, AKAZE generated almost no false inliers. For the Kilmoon Cross case, SIFT generated the largest group of false inliers, whereas AKAZE generated very few. With Boland's Mills SURF generated the worst results, whereas AKAZE generated the best, with the least number of false inliers. In all three cases, AKAZE performed extremely well.

The ratio of inliers to matches, as shown in Fig. 21, is an indication of detector efficiency, as generation of unusable keypoints is computationally expensive. AKAZE performed the best on all datasets in part by generating the fewest keypoints. SURF performed the worst on the Richview and Boland's Mills dataset, whereas ORB performed the worst on the Kilmoon Cross dataset.

The results in Fig. 22 compared the ratio of false matches to false inliers. An inlier is a match that has been validated by RANSAC. In this dataset, there should be no inliers, as there is no overlap between the images. Any matches that are found are false positives; the higher the percentage of inliers to matches, the higher the false positive rate.

5.4.3 *Timing results*

Although the reconstruction is done offline and real-time image processing is not required, the time taken to process a dataset does have an impact. In an exhaustive global comparison between images, the processing duration increases substantially with the number of images. Accordingly, an examination of keypoint generation and reconstruction times was conducted.

The time taken to process an image was consistent across all datasets, with SIFT requiring the longest time to generate the keypoints at 1.5 s per image. SURF and AKAZE were the second longest at 0.8 to 1 s to process an image. ORB and BRISK took the least amount of time, normally between 0.2 and 0.5 s per image, with very little variance as compared to the other operators. The only exception was in the Boland's Mills dataset, where AKAZE took longer (15 to 25 s) to generate the keypoints and had increased variance. This result is explained by the high number of keypoints generated by AKAZE on the Boland's Mills dataset (23,000 as opposed to an average of 4000).

The times were also recorded for each reconstruction using OpenMVG and are shown in the tables below. A number of interesting results were highlighted from this analysis. Most importantly, the matching time had a huge impact on the total time required for the reconstruction. Notably, different descriptors are used by each detector. SIFT uses a 128-byte vector, SURF uses a 64-byte vector, and AKAZE in this work uses a 64-byte vector, although it can also use a binary detector for matching. ORB and BRISK use binary descriptors with a length of 256 and 512 bits, respectively. Although matching binary descriptors should be faster as they only involve an exclusive or XOR comparison between bit strings, the binary descriptors required significantly more time (up to 240 times longer for ORB) due to the matching libraries being used. OpenMVG uses the fast library for approximate nearest neighbors[32] when comparing byte descriptors, whereas brute force matching is used for comparing binary strings. Approximate binary Hamming techniques, such as locality sensitive hashing[33] and hierarchical clustering[34] can be used to improve speed but were outside the scope of this work.

SIFT, SURF, and AKAZE processing times corresponded to the times required for inlier comparisons for detection, but the keypoint matching times differed (Tables 1–3). Those times correlated with the number of keypoints generated. SURF took the most time, followed by SIFT and AKAZE. ORB found the keypoints in the least amount of time, but the large number of keypoints found meant that the subsequent matching times were also increased. ORB took 240 times longer than SIFT to finish matching the Boland's Mills dataset where it found the most keypoints. This was exacerbated by the larger datasets and by the brute force matcher. Despite BRISK finding a similar number of keypoints to SIFT, SURF, and AKAZE, the matching time

Table 1 Richview results.

	Keypoint detection	Keypoint matching	Reconstruction	Total time
SIFT	1:32	2:54	1:25	5:53
SURF	0:51	3:44	2:15	6:51
ORB	0:17	51:03	3:12	54:34
AKAZE	0:56	1:20	0:56	3:13
BRISK	1:26	8:16	1:10	10:54

Table 2 Kilmoon Cross glasshouse results.

	Keypoint detection	Keypoint matching	Reconstruction	Total time
SIFT	2:13	5:11	0:41	8:07
SURF	1:18	8:08	1:01	10:29
ORB	0:17	32:30	0:46	33:34
AKAZE	1:18	1:31	0:22	3:11
BRISK	1:57	4:03	0:16	6:17

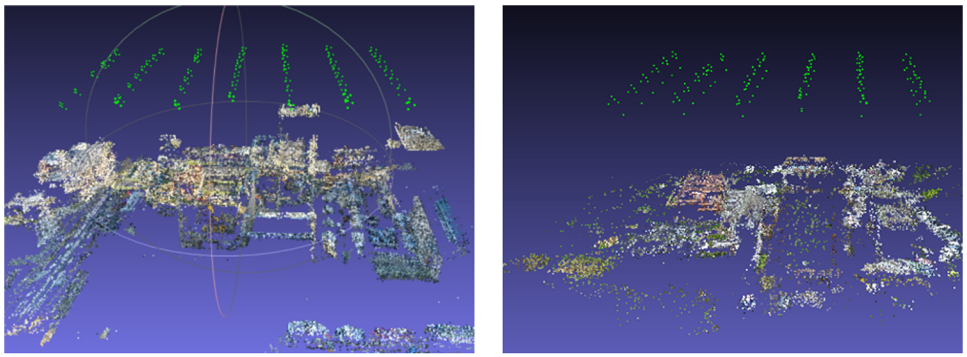
Table 3 Boland's Mills results.

	Keypoint detection	Keypoint matching	Reconstruction	Total time
SIFT	20:57	34:46	17:51	1:13:42
SURF	11:21	1:20:19	1:22:08	2:53:50
ORB	3:01	138:05:14	1:05:00	139:13:17
AKAZE	19:01	33:01	18:42	1:10:46
BRISK	3:25	9:01:41	8:15	9:13:23

was substantially longer, up to 15 times longer on the Boland's Mills dataset where it found the most keypoints. One other interesting point to note is that the time required for keypoint detection by BRISK should have been similar to ORB in OpenMVG, as the two methods had similar performance in the earlier inlier test. Both OpenMVG and the inlier tests used the same OpenCV implementation. So the reduced performance of BRISK may be a result of how BRISK is initialized in OpenMVG. As the required processing duration does not increase for a greater number of keypoints in the Boland's Mills dataset and, in fact, matches the processing speed of ORB for that dataset, the delay is believed to come from an initialization step in the BRISK detector.

5.4.4 *Reconstruction results*

Valid models were produced for all the detectors for the Richview and Boland's Mills datasets, but the reconstructions failed for the Kilmoon dataset. To give a measure of accuracy of the reconstructions, the models were overlayed with each other and aligned using iterative closest point (Fig. 23). This allowed for large artifacts generated by an individual detector to be easily observed. There were no large-scale artifact errors generated by the detectors. An additional step



(a) (b)

Fig. 23 The reconstructions for (a) Boland's Mill and (b) Richview aligned on top of each other. The wall ratios were found to be consistent across all reconstructions. The green dots are the estimated camera poses generated during the reconstruction.

of calculating the wall ratios of two main buildings from each dataset was conducted to measure if the resulting model contained distortions not found in the original scene. The wall ratios were found to be consistent across all reconstructions.

Although metrics, such as the number of keypoints generated and the number of inliers, are traditionally used to evaluate the efficacy of a feature detector, arguably a more accurate measure is how many of these keypoints are ultimately part of the sparsely reconstructed cloud. Herein, these were evaluated with respect to the number of poses, tracks, and keypoints. The pose count is the number of camera positions (rotation and location in 3-D space) that are successfully calculated from the 2-D images; the track is the position of a characteristic point that is corresponding across multiple images; and the number of points is the sparse point cloud result. The root-mean-square error (RMSE) of the pixels is used to indicate the residual error when aligning the camera poses. The results for the three datasets are shown in Tables 4–6.

The reconstruction results for Richview (Table 4) show that all of the keypoint detectors managed to calculate the camera position for all of the images. Although the feature detector ranking is the same as the keypoint results and the inlier results, the amount that was included in the final point cloud reduced the scale of the difference. For example, ORB generated an average of 15,000 keypoints and 300 inliers per image, but it only resulted in 7809 points in the final model. In comparison, SIFT generated only 5000 keypoints and 80 inliers per image, but there were 4828 points in the final point cloud. Additionally, the RMSE was much lower for the SIFT cloud, which would indicate a higher accuracy. The results imply that while more keypoints result in a denser point cloud, the cost in terms of generating and matching the keypoints may outweigh the benefit.

The results for the Kilmoon Cross glasshouse are shown in Table 5. This dataset proved problematic for all the detectors, with over half the images not ultimately positioned in the final model. SIFT managed to pose the most cameras at 32 locations. The detector ranking matched with the inlier results in Fig. 17, but again, the differences were massively reduced, and the residual error was much lower.

Table 4 Reconstruction results for Richview.

	Pose count	Tracks	RMSE	Number of points
SIFT	163/163	26,240	0.2857	4828
SURF	163/163	31,775	0.4606	5950
ORB	163/163	38,750	0.4767	7809
AKAZE	163/163	16,739	0.5021	4386
BRISK	163/163	20,917	0.4455	4921

Table 5 Reconstruction results for Kilmoon glasshouse.

	Pose count	Tracks	RMSE	Number of points
SIFT	32/82	2684	0.265	2716
SURF	27/82	3001	0.4919	3028
ORB	27/82	3178	0.4044	3205
AKAZE	28/82	1320	0.4699	1348
BRISK	20/82	1346	0.3686	1366

Table 6 Reconstruction results for Boland's Mills.

	Pose count	Tracks	RMSE	Number of points
SIFT	52/52	48,037	0.6282	48,089
SURF	52/52	75,220	0.7007	75,272
ORB	52/52	65,634	0.7019	65,686
AKAZE	52/52	63,106	0.7609	63,158
BRISK	52/52	38,979	0.7296	39,031

For Boland's Mills (Table 6), all of the detectors managed to calculate the camera positions. The results differed from the keypoint and inlier metrics and more closely correlated with the ratio test metric results. The ratio test is simply a heuristic for filtering outliers from a dataset and, as seen in the previous results, may not indicate the quality of the final reconstruction. All approaches generated a large number of points, but SURF performed the best overall, closely followed by ORB and AKAZE. SIFT and BRISK performed the worst. The residual errors were also similar with SIFT producing the smallest RMSE.

6 Discussion

The focus of feature detectors in recent years has been to increase the speed of the computation and increase the number of inliers for rapid matching in robotic and real-time vision applications. Importantly, the requirements of SFM are different in that usable keypoints that can be tracked across multiple images are more useful in reconstruction. Additionally, quickly generating more keypoints of lower quality actually hinders reconstruction when the descriptors for the dataset must be globally matched. Furthermore, the metrics normally used for evaluating detectors such as inliers had little effect on the final reconstruction, with most of the generated keypoints discarded during bundle adjustment. The final output only used a small percentage of the generated keypoints in the final model.

Two interesting results emerged from the above experiments: (1) having a large number of inliers does not necessarily translate into more points in the final point cloud reconstruction (despite a traditional reliance on this as a metric for detectors), thereby demonstrating that the cost of comparing a large quantity of redundant keypoints may outweigh the advantage of having the extra points and (2) despite the explosion of keypoint detectors and descriptors, none of them led to a substantial improvement in the quality of SFM reconstruction. In fact, SIFT, the oldest detector tested in this paper was consistently (at least marginally) better at estimating camera poses during the final reconstruction stage.

An overview of the performance of the operators is given in Fig. 24. The comparison of accuracy versus time taken is used to give an idea of the relative performance of the operators across the three cases. The descriptors are indicated by the shape (triangles equal SIFT, square

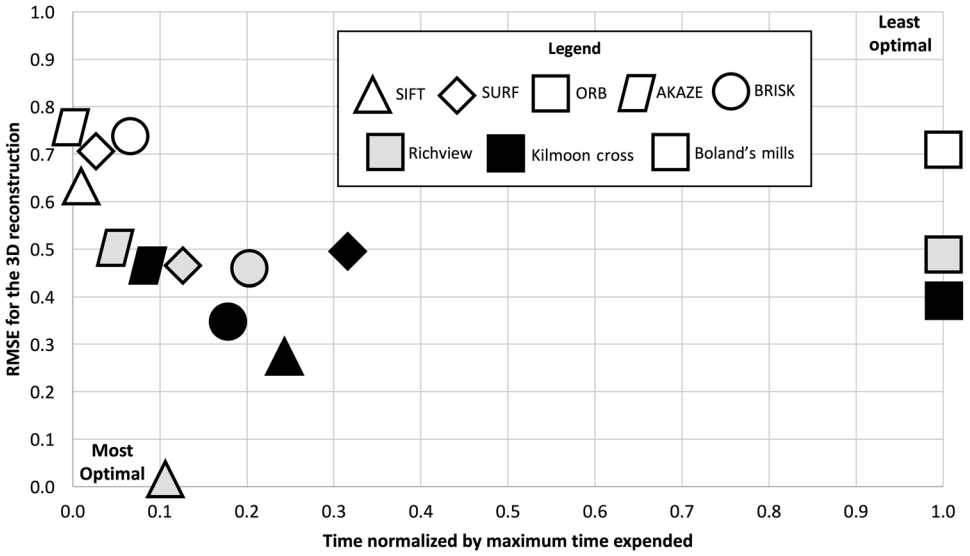


Fig. 24 A performance overview of the keypoint detector.

equals AKAZE, etc.) and the datasets are indicated by text, border, and fill (gray equals Richview, white equals Boland's, and Kilmoon equals black fill). The accuracy is deduced from the reconstruction results, whereas the time is the total time taken for the reconstruction and normalized by the longest duration.

7 Conclusions

This work compared five feature descriptors (SIFT, SURF, ORB, AKAZE, and BRISK) on three different aerial datasets captured by UAV with the intention of examining if there was a performance improvement in reconstruction quality. The datasets were gathered at low altitude by UAVs with no rotational or scale change and with an 80% overlap. The results showed that, although some feature descriptors generated more keypoints and more inliers, they did not necessarily lead to a better quality 3-D reconstruction. The number of points that passed inlier evaluation was dramatically reduced by the bundle adjustment step from a factor of 10 to a factor of 2 in the output cloud. The additional time spent evaluating the matches also massively increased the processing time, which negated the benefit of having more inliers. Examining the final output clouds showed that despite the huge variance in keypoint generation and inlier matching of the five detectors, there was only marginal difference in the final reconstruction with SIFT having the smallest residual error across all datasets.

Acknowledgments

This work was funded Science Foundation Ireland Grant No. 13/TIDA/1274, Irish Research Council for Science Engineering and Technology (IRCSET) Doctoral Grant No. GOIPG/2015/3003, IRCSET Post-Doctoral Grant No. GOIPD/2015/125, and by the European Commission Grant No. ERC StG 2012-307836-RETURN. Additional funding for this project was provided in part by Geological Survey of Ireland Grant No. SC2015_Laefer.

References

1. A. Rango et al., "Unmanned aerial vehicle-based remote sensing for rangeland assessment, monitoring, and management," *J. Appl. Remote Sens.* **3**, 033542 (2009).
2. S. Chen, D. Laefer, and E. Mangina, "State of technology review of civilian UAVs," *Recent Pat. Eng.* **10** (3), 160–174 (2016).
3. D. L. Milgram, "Computer methods for creating photomosaics," *IEEE Trans. Comput.* **C-24**, 1113–1119 (1975).

4. J. Byrne and D. Laefer, "Variables effecting photomosaic reconstruction and orthorectification from aerial survey datasets," 2016, https://arxiv.org/submit/1705751/view (20 November 2016).
5. T. Jebara, A. Azarbayenjani, and A. Pentland, "3D structure from 2D motion," *IEEE Signal Process Mag.* **16**(3), 66–84 (1999).
6. G. Bradski, "The opencv library," *Doct. Dobbs J.* **25**(11), 120–126 (2000).
7. P. Moulon, P. Monasse, and R. Marlet, "Global fusion of relative motions for robust, accurate and scalable structure from motion," in *Proc. of the IEEE Int. Conf. on Computer Vision*, Sydney, pp. 3248–3255 (2013).
8. D. Lowe, "Object recognition from local scale-invariant features," in *7th IEEE Int. Conf. on Computer Vision*, Vol. 2, p. 1150 (1999).
9. D. Marr and E. Hildreth, "Theory of edge detection," *Proc. R. Soc. B: Biol. Sci.* **207**(1167), 187–217 (1980).
10. H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: speeded up robust features," in *European Conf. on Computer Vision*, pp. 404–417 (2006).
11. E. Rublee et al., "ORB: an efficient alternative to SIFT or SURF," in *European Conf. on Computer Vision*, pp. 2564–2571 (2011).
12. E. Rosten, R. Porter, and T. Drummond, "Faster and better: a machine learning approach to corner detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(1), 105–119 (2010).
13. M. Calonder et al., "BRIEF: binary robust independent elementary features," in *European Conf. on Computer Vision*, pp. 778–792 (2010).
14. C. Harris, "A combined corner and edge detector," in *Proc. of the Fourth Alvey Vision Conf.*, pp. 147–151 (1988).
15. P. L. Rosin, "Measuring corner properties," *Comput. Vision Image Understanding* **73**(2), 291–307 (1999).
16. P. F. Alcantarilla, J. Nuevo, and A. Bartoli, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," in *British Machine Vision Conf. (BMVC)*, Bristol, United Kingdom (2013).
17. W. H. Press et al., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, New York (2007).
18. H. Scharr, "Optimal operators in image processing," Doctoral dissertation, Rupertus Carola University, Heidelberg, Germany (2000).
19. S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *Int. Conf. on Computer Vision*, pp. 2548–2555 (2011).
20. E. Mair et al., "Adaptive and generic corner detection based on the accelerated segment test," in *European Conf. on Computer Vision*, Berlin, pp. 183–196 (2010).
21. DJI, "Phantom 3 professional product information," 2016, http://www.dji.com/product/phantom-3-pro/info (1 December 2016).
22. Pix4D, "Pix4D capture mapper," Pix4D, 2011, www.pix4d.com (3 November 2016).
23. J. Byrne et al., "3D reconstructions using unstabilised video footage from an unmanned aerial system," *J. Imaging* **3**(2), 15 (2017).
24. D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Pearson, Upper Saddle River, New Jersey (2003).
25. M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM* **24**(6), 381–395 (1981).
26. B. Triggs et al., "Bundle adjustment—a modern synthesis," in *Vision Algorithms: Theory and Practice*, pp. 298–372, Springer-Verlag, Berlin, Heidelberg (1999).
27. Y. Furukawa et al., "Towards internet-scale multi-view stereo," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR '10)* (2010).
28. K. Lenc, V. Gulshan, and A. Vedaldi, "VLBenchmarks," 2012, http://www.vlfeat.org/benchmarks (14 November 2016).
29. S. Gottwald, *The VNR Concise Encyclopedia of Mathematics*, 2nd ed., Springer, New York (2012).
30. P. Moulon and P. Monasse, "Unordered feature tracking made fast and easy," in *Conf. on Visual Media Production (CVMP)* (2012).

31. L. Moisan, P. Moulon, and P. Monasse, "Automatic homographic registration of a pair of images, with a contrario elimination of outliers," *Image Process. On Line* **2**, 56–73 (2012).
32. M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISAPP Int. Conf. on Computer Vision Theory and Applications*, Lisbon, Portugal, pp. 331–340 (2009).
33. A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Int. Conf. on Very Large Data Bases*, Edinburgh, Scotland, pp. 518–529 (1999).
34. S. Johnson, "Hierarchical clustering schemes," *Psychometrika* **32**, 241–254 (1967).

Jonathan Byrne received his PhD in evolutionary design optimization in 2012, using machine learning techniques to optimize the design of bridges, electricity pylons, and aircraft. He is a computer vision researcher working with Intel on the Movidius Myriad computer vision chip. Previously, he worked as a postdoctoral researcher at the Urban Modelling Group and was a cofounder of U3D. He is currently developing design optimization techniques that take advantage of the unique properties of the additive manufacturing process.

Debra F. Laefer received her degrees from the University of Illinois at Urban-Champaign (PhD), New York University (MS), and Columbia University (BA, BS). She is a professor of urban informatics at the Center for Urban Science and Progress, New York University. She is also an adjunct faculty member in civil engineering at the University College Dublin.

Evan O'Keeffe is an Irish Research Council funded PhD student at the University College Dublin. He received his honors BSc degree in computer science from UCD and MSc degree in enhancing UAV photogrammetry from UCD. His PhD project "Multi-mode control of multiple, synchronised unmanned aerial vehicles (UAV)" looks at using smart radio systems to improve visual image feedback during UAV flight operations. His areas of interest include image processing, three-dimensional reconstruction, software defined radio, and cognitive radio.