

Affine Invariant OpenCV Descriptors and the Effects on Aerial Photogrammetry

Evan O’Keeffe ¹, Debra F. Laefer ^{1,2}, and Eleni Mangina ³

¹ Center for Urban Science + Progress, Tandon School of Engineering, New York University, 370 Jay Street, 13th Floor, Brooklyn, NY 11201, USA

² Dept. of Civil, Urban, and Environmental Engineering, Tandon School of Engineering, New York University, Brooklyn, NY, USA

³ School of Computer Science, University College Dublin, Belfield, Dublin, 4, Ireland

Keywords: OpenCV, OpenMVG, Image descriptor, Image matching, Affine invariance, Photogrammetry.

Abstract

Robust feature descriptors are necessary for computer vision applications such as image matching, photogrammetric three-dimensional (3D) reconstructions, and simultaneous localisation and mapping (SLAM). While most state-of-the-art feature descriptors are invariant to image transformations (such as translation, rotation, and scale) the majority lack stability in tracking points over large 3D perspective transformations. One successful method to solving these large perspective changes is by simulating affine tilts on the latitude and longitude axes of an image. These simulated tilts create greater invariance to changes in 3D perspective. To demonstrate the widespread efficacy of this approach, this paper applies affine simulation to seven state-of-the-art descriptors in OpenCV and to two of the enhanced OpenCV descriptors in OpenMVG to 5 datasets that all include large affine conditions. Invariant descriptors were able to achieve up to a three-fold increase in feature tracking.

1. Introduction

Feature detectors and descriptors are the cornerstones of image processing and analysis. Feature detectors find points in an image, while feature descriptors reliably enable similar points in two images to be found. Local image features have proven to be useful in a range of activities including image matching (Remondino et al., 2014), object detection (Wohlhart and Lepetit, 2015), simultaneous localisation and mapping (SLAM) (Cvišić et al., 2018), augmented reality (Neumann and You, 1999), and three-dimensional (3D) Structure from Motion (SfM) reconstructions (Shibata et al., 2015). Key points are found by looking at unique image features such as edges, lines, blobs, regions or ridges. Each extracted key point has a two-dimensional (2D) position in the image and may provide coverage information (i.e. pixel or patch of pixels). To be able to match these key points in other images, there must be a way of distinguishing them from each other. For this, a feature descriptor is needed.

Feature descriptors must be able to identify key points regardless of a key point’s location with respect to changes in translation, rotation, scale, illumination, and 3D perspective view. One way is by using the intensity of the direction of the most pronounced orientation. Another is by concatenating responses from operations, such as through the use of 2D Haar wavelets (Porwik and Lisowska, 2004).

Effective key point detectors and descriptors must be able to match points in different images with potentially large changes in translation, rotation, scale, 3D perspective changes (i.e. affine view change), and light variance. The ability to do so is essential for operations such as stable and robust object detection, robotic collision avoidance, and structure from motion reconstruction. Feature matching in key point detector algorithms generally requires at least three key points to be matched and may require multiple sets of three matched key points. While the goal is for all tasks to be completed as quickly and efficiently as possible, some tasks are more lenient to softer time goals than others. For example, searching for duplicate images can take a long time, but having an algorithm that finds a greater number of duplicates may be more valuable than one that can

more quickly find only some of them. Such trade-offs are often application-specific.

Common feature detectors and descriptors include binary robust independent elementary features (BRIEF), oriented FAST and rotated BRIEF (ORB), binary robust invariant scalable keypoints (BRISK), scale invariant feature transform (SIFT), speeded up robust feature (SURF), accelerated kaze (AKAZE), and learned arrangement of three patches (LATCH). Table 1 describes these feature descriptors as implemented in the OpenCV library (Bradski, 2000) and their invariance to various image transformations. These include (1) AKAZE (Alcantarilla et al., 2013), which works by using finite element diffusion to accelerate the Gaussian scale estimation in conjunction with either a floating-point descriptor or a modified local difference binary descriptor for improved detection and enhanced matching performance and (2) LATCH (Levi and Hassner, 2016), which is built on local binary patterns using triplets of mini-patches to create a descriptor for 512 bits and has been shown to have very good performance compared to SIFT (Lowe, 2004) and AKAZE (Alcantarilla et al., 2013).

Name	Type	Translation	Rotation	Scale	Affine
BRIEF	Bin	Yes	Yes	-	-
ORB	Bin	Yes	Yes	semi	-
BRISK	Bin	Yes	Yes	-	-
SIFT	UInt	Yes	Yes	Yes	< 50°
SURF	Float	Yes	Yes	Yes	< 30°
AKAZE	Bin	Yes	Yes	Yes	-
LATCH	Bin	Yes	Yes	Yes	-

Table 1. OpenCV Descriptor Invariance.

The primary problems that most detectors and descriptors solve are image recognition, object detection, and camera motion estimation (e.g. SLAM) (Kuai et al., 2010). These are typically applied across very large data sets, which is why a detector and descriptor that minimises computational time and memory is attractive. For example, with SLAM (Wang et al., 2012), if the key point matching cannot operate with sufficient rapidity, a robot’s speed and viability in terms of positional tracking and its battery life may preclude its real-world deployment.

Critically, quality photogrammetry processing requires a descriptor as robust as possible for the aforementioned transformations. As will be demonstrated, current descriptors cannot track features over affine transforms greater than 30° , this paper aims to show how current feature descriptors can be augmented to track across 3D perspective changes through the introduction of affine simulation. The specific contributions of this paper include (1) a quantitative comparison of the affine versus non-affine versions of the current state-of-the-art OpenCV (Bradski, 2000) implementations of feature extraction and description algorithms, when applied to benchmark data set (Mikolajczyk and Schmid, 2004) and (2) an extensive reporting on 3D reconstructions resulting from affine and non-affine versions of the SIFT and AKAZE feature descriptors. This latter contribution was achieved using the OpenMVG framework (Moulon et al., 2016) with MVE (Fuhrmann et al., 2014) as it enables the introduction of OpenCV descriptors, where built-in descriptors (as with SIFT & AKAZE) were not otherwise available. The use of OpenMVG also allows the interrogation of the system to see the performance of the descriptors in full.

2. Related Work

Managing image feature tracking requires robustness to multiple image transformations. Various strategies have been devised to gain this robustness. This section aims to present some of the most notable work to date to identify remaining limitations.

Changes in translation and rotation have often been solved using normalisation techniques, while scale changes have been solved using Laplacian of Gaussian (LoG) (Sotak and Boyer, 1989), degrees of Gaussian (DoG) (Lowe, 2004), and fast explicit diffusion (FED) (Alcantarilla et al., 2013). Yu et al. (2011) solved the problem of full, 3D perspective view transformations in much the same way that scale issues were solved using DoG pyramids (Lowe, 2004), which simulate the longitude and latitude tilts through an affine transform.

Work by Yu and Morel (2011) showed that a general affine simulation method can be applied to feature descriptors, to enhance the tracking across images with large changes in 3D perspective view. Previous work in this area demonstrated that descriptors such as SIFT (Yu and Morel, 2011), SURF (Pang et al., 2012), and ORB (Codreanu, 2014) significantly increased tracking, with respect to 3D perspective view changes.

In the approach by Yu and Morel, each image was transformed by simulating all affine distortions based on the longitude ϕ and latitude θ tilts with parameter $t = \frac{1}{\cos \theta}$. This was accomplished by directional t-subsampling, which required the pre-application of anti-aliasing filters in the x-direction. Gaussian blurring was then used with a standard deviation $c\sqrt{(t^2 - 1)}$ letting $c = 0.8$, as was previously used (Yu and Morel, 2011). With $t = 7$, this method has been shown to be fully affine invariant (Yu and Morel, 2011). The next step in that method involved computing the detector and descriptor for each of the simulated images. This means that any detector and descriptor could be used in place of another.

Despite the success of these affine simulations on SIFT (Yu and Morel, 2011) and SURF (Pang et al., 2012), documented experimentation on other openly available descriptors has not occurred. Therefore, the usefulness of the approach is not known for many of the other openly available descriptors.

Subsequently, Yu et al. (2018) investigated methods to improve feature matching for oblique aerial imagery using ASIFT. They modified the feature matching method by correcting feature points using a weighted least squares matching method. In that approach, mismatched points were discarded, and a local transform model (or homography) was calculated using an adaptive normalised cross-correlation algorithm. This increased tracking was achieved by making the descriptor more robust to affine transformations and resulted in an average increase of 26.75% over SIFT using Locally Linear Transforms.

Previously, Zheng et al. (2014) directly compared multiple variations of SIFT (i.e. PCA-SIFT (Ke and Sukthankar, 2004), GSIFT (Lodha and Xiao, 2006), CSIFT (Abdel-Hakim and Farag, 2006), and ASIFT (Yu and Morel, 2011)) for matching drone-based images with satellite images. They showed ASIFT as effective for tracking changes in rotation from 0° to 310° ; the focus of the paper was modifying ASIFT with an elliptic pie chart sampling grid to reduce computational complexity and reduce matching error rates. They used a synthetic unmanned aerial vehicle (UAV) image set retrieved from Google Earth in which the roll angle was fixed as 45° and rotation angle changed from 0° to 315° with an interval of 45° , for satellite imagery of Baidu. They were able to decrease the mean time from 18.27 seconds for ASIFT to 0.85 seconds for their proposed method and increased matching efficiency from ASIFT's 5% to 13% in their proposed method. The regular SIFT had a mean time of 8.53 seconds and matching efficiency of 0.8%. This paper showed that ASIFT with a sampling grid was quite effective for improving matching time and performance in rectifying UAV to satellite imagery.

In summary, having affine invariant feature descriptors can be highly effective in increasing feature tracking in image processing with respect to 3D perspective view changes. While the descriptors SIFT, SURF, and ORB have already been tested, affine simulations have not been applied to many of the other open source descriptors. This paper aims to do so in direct comparison with some previously tested descriptors and critically, to investigate the impact of applying an affine approach on 3D reconstructions derived from these processed data sets.

3. Methodology

This research undertakes two groups of experiments. The first benchmarks OpenCV descriptors against their affine invariant counterparts using benchmark data sets from Mikolajczyk and Schmid (Mikolajczyk and Schmid, 2004). Those widely used collections of images provide data appropriate for investigating multiple image processing problems (e.g. blur, viewpoint change, zoom plus rotation, light, and JPEG compression). Each of those datasets contain 6 images of increasing 3D perspective view change (see Figure 1 where (a) is the Wall and (b) is the Graffiti), along with 5 homography matrices. Each homography matrix is the relationship transformation from the first image to the second, third, fourth, fifth, and sixth images. The homography matrix enables transformation of points from image 1 to image N. Having this enables the tester to validate the transformations generated from matching the descriptors from image 1 to image N. Since the homography (i.e. the mapping of one image to another) matrices are known, each descriptor match can be validated to find the number of inliers (correct matches) versus outliers (false matches). This allows direct comparison of which feature descriptor matches are true pos-

itive matches versus false matches that can arise from duplicate patterns in the image(s).

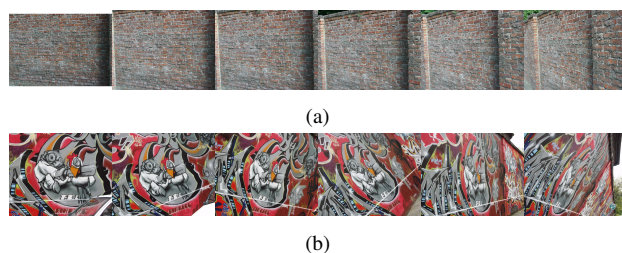


Figure 1. Affine Images from Mikolajczyk and Schmid (Mikolajczyk and Schmid, 2004): (a) Wall images. (b) Graffiti images.

The viewpoint change was created from a front-parallel view to one with significant foreshortening at approximately 60° to the camera. All images are of medium resolution of approximately 800×640 pixels. Having known homographies also allows quantification of the execution time of the feature detection, description, and matching stages. The ability to compare different stages of the processes provides further insight as to how future optimization may be achieved.

A wrapping function was created that allows a programmer to specify an OpenCV detector and descriptor to use with the affine algorithm created by Yu and Morel (2011). Through this, the image detectors and descriptors were used as normal OpenCV functions for image feature detection, description and matching. This was achieved using a C++ template that employs OpenMP (OpenMP ARB, 2021) to parallelize the computation.

A program was created using this wrapping function to test the different feature detector and descriptors with the Wall and Graffiti datasets. This enabled the calling of specific OpenCV feature detectors and descriptors, in both affine and non-affine versions.

For each detector and descriptor, the selected matching ratio used was taken from the original citation. The use of matching ratios allowed for exact value comparison against the tested affine variants. The reported run time includes only image feature detection, feature description, and matching. Image loading time was not considered. The list of feature descriptors to be tested can be seen in Table 2.

Descriptor	Experiment 1	Experiment 2
STAR-BRIEF	X	
ORB	X	
BRISK	X	
SIFT	X	X
SURF	X	
AKAZE	X	X
LATCH	X	

Table 2. OpenCV Descriptors Tested in Each Experiment
Experiment 1 assessed impact of affine algorithm (Yu and Morel, 2011) on Wall and Graffiti benchmark images
Experiment 2 assessed impact of affine algorithm (Yu and Morel, 2011) on 3D reconstructions.

Experiment 1 was conducted by (1) computing the regular detectors and descriptors on each of the images, (2) matching the descriptors in the images, and (3) computing the inlier ratios.

Then the tests were rerun using the detectors and descriptors with the affine simulations. In both cases, the results were piped to a text file.

Experiment 2 was designed to explicitly determine the impact the affine descriptors have on 3D reconstructions. This second experiment used OpenMVG (Moulon et al., 2016) with OpenCV versions of the descriptors in Table 2. These descriptors were used with the wrapper used in Experiment 1 to generate the affine versions of the descriptors. These were applied in three case studies with various image capture perspectives: oblique (Figure 2), nadir (Figure 2), and a vertical Boustrophedon in which a drone flew up and down in alternating columns. In OpenMVG, two binaries were generated: one used the regular OpenCV descriptors and the other used the modified affine versions. The use of OpenMVG enabled the generation of a report at the end of the reconstruction process that could be used to compare the approaches with respect to Root Mean Square Error (RMSE), track length (min, mean, and max times features that are found in other images), and inlier matching counts. This allows for a more robust evaluation of feature descriptor effectiveness in the case studies. Each of these was tested for tracking ability with and without the affine transformation and for how the affine transformation affected photogrammetric 3D reconstructions.

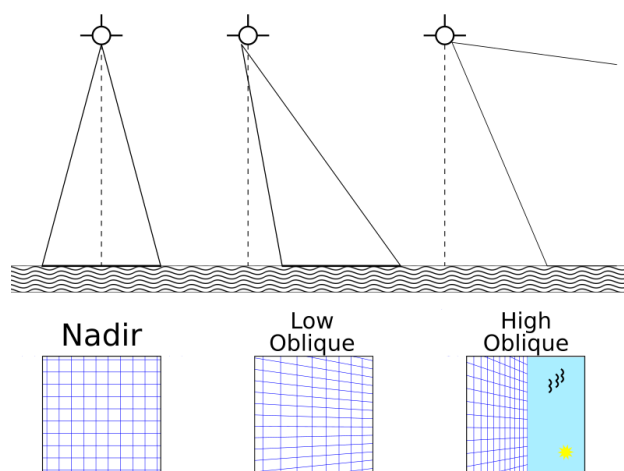


Figure 2. Oblique vs. Nadir Imagery.

To investigate the second problem, other real world image data was used. While standard data sets that evaluate translation, rotation, and scale are available (mostly for nadir imagery), none exists for large 3D perspective view points that contain known homographies against which to validate. Three such datasets are, thus, introduced herein to evaluate the usefulness of the affine enhanced feature descriptors for 3D reconstructive purposes.

First the UCD Richview data from (Byrne et al., 2017) (Figure 3), as the dataset has documented outcomes for a number of feature descriptors and their outcomes. The imagery has an 80% overlap at an oblique angle, for which the original descriptors should produce some comparative values.

The second dataset is a nadir capture of Bandon town in Cork Ireland (Figure 4). This dataset features many buildings located on uneven terrain. The dataset has again an 80% overlap between images with a nadir perspective, for which the original descriptors should produce some comparative values.



Figure 3. Sample Image from the UCD Richview Dataset.



Figure 4. Sample Image from the Bandon Town Dataset.

The last dataset chosen is from (Chen et al., 2018). The images are of the Boyne Viaduct, a 30-metre-high railway bridge, that crosses the River Boyne in Drogheda (Figure 5). The imagery contains an overlap of at least 60% at oblique and nadir perspectives, for which the original descriptors should produce some comparative values.



Figure 5. Sample Image from the Boyne Bridge Dataset.

Affine invariant versions of the SIFT and AKAZE descriptors were tested against the non-affine simulated SIFT and AKAZE descriptors using the three data sets mentioned above. For each, two key metrics were recorded: (1) image feature tracking over the data set and (2) RMSE of the SfM reconstruction. After using OpenMVG to reconstruct a sparse point cloud, then MVE (Fuhrmann et al., 2014) was used to generate a dense point cloud using the openMVG_main_openMVG2MVE2, dmrecon and scene2pset programs. This allowed for a qualitative comparison of the point clouds generated from the regular versus affine invariant feature descriptors. The workflow is illustrated in Figure 6.

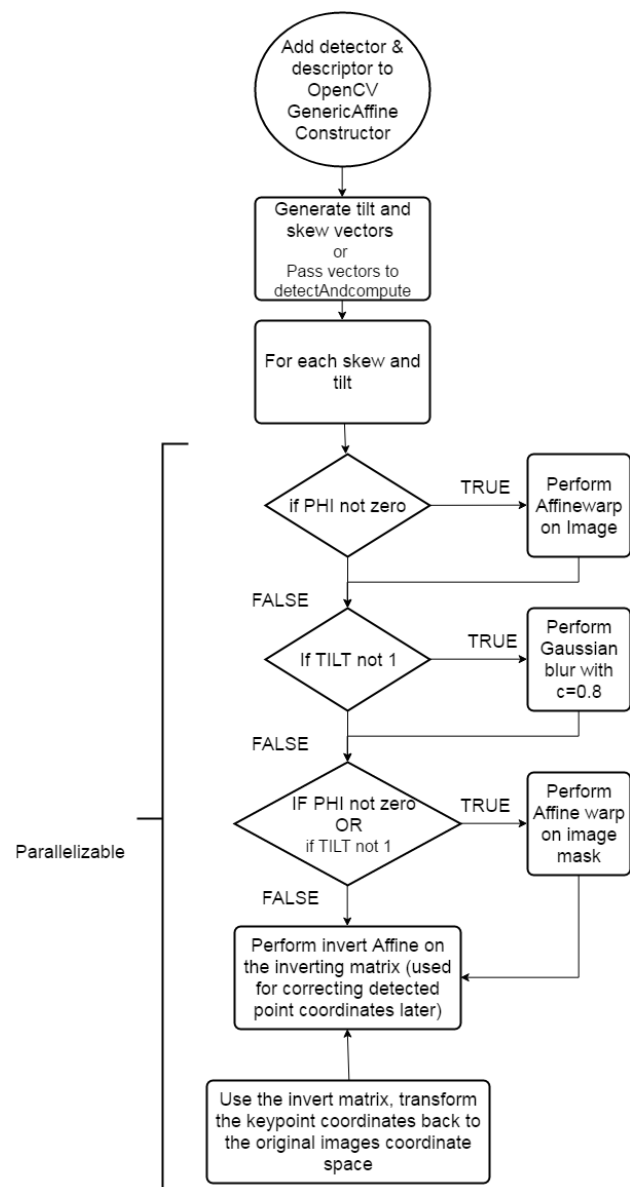


Figure 6. Affine Simulation Workflow.

4. Results

This section details the results produced using the methodology previously described for the two experiments. The first compares OpenCV descriptors with and without affine enhancement on two benchmark datasets. The second compares regular

versus affine versions of SIFT and AKAZE with the OpenMVG 3D SfM reconstruction framework and MVE (Fuhrmann et al., 2014) on data real world scenes. These results were generated on a dual-core i5 @3GHz machine with 16GB of RAM running Ubuntu 16.04 LTS with OpenCV 3.1.

4.1 Testing the OpenCV Feature Descriptor Tracking

The primary datasets for this experiment were from Mikolajczyk and Schmid (Mikolajczyk and Schmid, 2004) specifically the datasets Wall (Figure 1a) and Graffiti (Figure 1b).

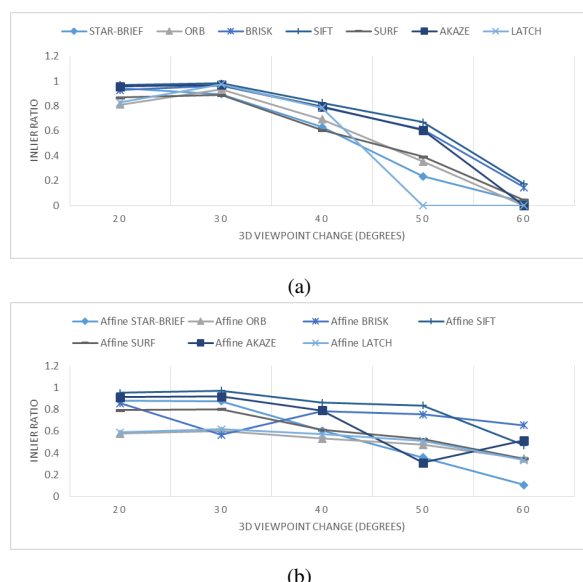


Figure 7. Wall Dataset Results: (a) Regular descriptors. (b) Affine descriptors.

As visible in Figure 7a, all descriptors were able to keep their inlier ratios above 0.5 when the viewpoint change was limited to 40°. Beyond that angle, all the descriptors struggled to match keypoints, and at 50° the LATCH descriptor completely failed (Figure 7a). In contrast, the affine versions were able to maintain viewpoint tracking even at viewpoint changes beyond 60°. Affine BRISK and Affine AKAZE had the best tracking inlier ratios at the 60° mark, with Affine BRISK requiring less computational time (having the smallest slow down – only 36% versus the normal BRISK).

The Graffiti dataset was significantly more complex due to its addition of viewpoint change and camera rotation. Consequently, earlier failure of all descriptors was expected (Figure 8a). In this case, by the 40° degree viewpoint change, BRIEF and LATCH both failed completely, with STAR-BRIEF performing badly from the beginning. With this dataset, almost all the descriptors completely failed tracking by the 50° viewpoint marker. Even BRISK, which was the best through the early parts of the test completely failed at 60°. The addition of Affine helped most of the techniques. The exceptions were STAR-BRIEF, where the impact was marginal and BRISK where the outputs were inconsistent (Figure 8b). Most importantly, however, the addition of Affine enabled the original techniques (except for BRIEF) to continue to work with as much as a 60° viewpoint change.

While these results showed that the affine simulations made the descriptors more robust, this was not without a cost. The graphs in Figure 9 show the inlier ratios with the respective slow down

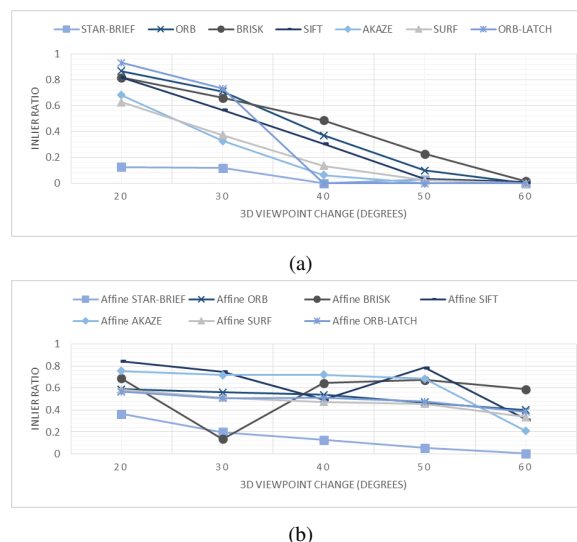


Figure 8. Graffiti Dataset Results: (a) Regular descriptors. (b) Affine descriptors.

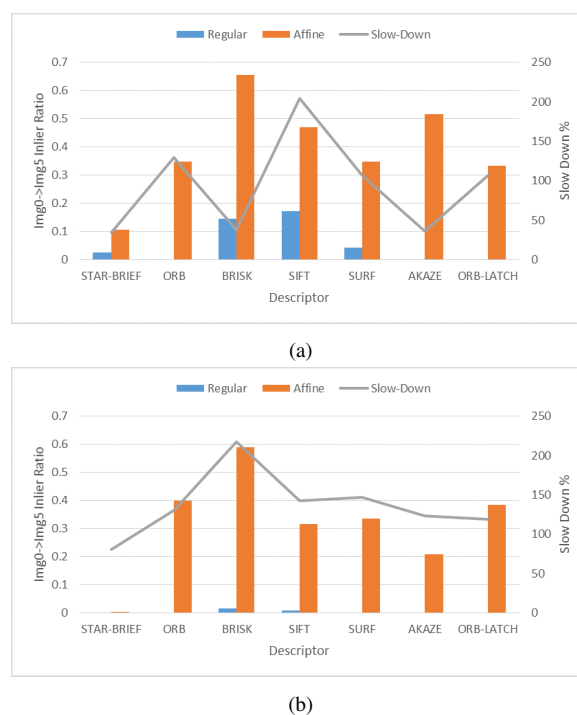


Figure 9. Slowdown for the Wall & Graffiti Datasets: (a) Wall dataset. (b) Graffiti dataset.

for each descriptor for the Wall and Graffiti datasets at the 60° viewpoint change in (Figure 9a and Figure 9b, respectively). For the Wall dataset, there was an extremely poor correlation between increases in the inlier ratio and the slow down. In fact, for BRISK and AKAZE where the improvement was the greatest (compared to the non-affine versions), the slowdown was the least. The performance was, however, much more consistent for the more complex Graffiti dataset, where the impact of affine was even more important in terms of achieving a usable number of inliers (i.e. on average an inlier ratio of 0.3195 with the affine compared to only 0.0037 without) with an average 137% slowdown.

Each of the normal versions of the descriptors had a relatively

low number of feature points, while the affine enhanced versions typically had 16 times as many keypoints and descriptors. This had a cascading effect in the inlier ratio averages for the affine descriptors, as they had to match more feature points versus the normal descriptors.

One item that was investigated was whether artificially increasing the keypoints extracted using the regular descriptors might improve the inlier matching ratio. As visible in Table 3, artificially increasing keypoints had no impact on the inlier matching ratio. When the affine enhanced descriptors were used, a more positive set of results were obtained. Specifically, at the largest affine transformation (i.e. matching feature keypoints in image 0 to image 5) meaningful inlier ratios were achieved.

Name	Regular Keypoints	Affine Keypoints	Matches	Regular Inlier Ratio	Affine Inlier Ratio
BRISK	21,930	15,672	593	0.000	0.655
AKAZE	4,910	4,284	335	0.000	0.516
SIFT	10,357	11,046	591	0.002	0.470
ORB	9,000	9,000	216	0.000	0.348
SURF	10,418	7,850	409	0.000	0.347
ORB-LATCH	23,461	22,879	141	0.000	0.333
BRIEF	672	447	91	0.000	0.105

Table 3. Increased Number of Keypoints Descriptor Tests.

These results show that overall there was a vast improvement in matching performance (e.g. BRISK went from 0% inliers to 0.58% inlier ratio for the Graffiti dataset). To verify that this was not caused by a sheer increase in feature points, a further test was conducted with the feature detectors and descriptors capturing substantially more points either by increasing the number in the OpenCV constructor or by running the feature descriptor on a single image multiple times. The results are shown in Table 3.

The results demonstrated that the affine simulations improved the feature descriptors substantially (Figure 9). This could be in part due to the nature of the binary matching operations in descriptors such as ORB, which are very fast and efficient, compared to the histogram matching methods in descriptors such as SIFT and SURF. The descriptors that seemed to benefit the most whilst operating the fastest were ORB, BRISK, SIFT, SURF, and AKAZE. In particular ORB, BRISK and AKAZE seemed to improved the most, while minimising resource usage. Arguably, despite the increased processing time, ORB, BRISK and AKAZE would be the most beneficial in terms of trade off between accuracy and processing time (Figure 9).

4.2 Photogrammetry Results Comparing Affine Invariant Descriptors

4.2.1 Dataset 1: Oblique Images - UCD Richview This test involved comparing SIFT versus ASIFT for the Richview aerial dataset (Figure 12). The results in Table 4 indicate that for the oblique dataset, there was a large improvement in feature tracking. The affine version yielded 135 viewpoints from the 149 that were available, versus the 80 viewpoints that the original SIFT was able to reconstruct. The number of multiview correspondences increased from 7,320 for the original SIFT to 89,274 with ASIFT, along with the number of residuals increasing from 35,000 in the original to 339,000 points in the affine version. The scene RMSE increased by 0.09 from the original to the affine; this is probably due to the increased number

of points matched against in the affine version. These results show that in oblique imagery, the affine invariant descriptor is able to reconstruct significantly more of the scene than the normal SIFT descriptor.

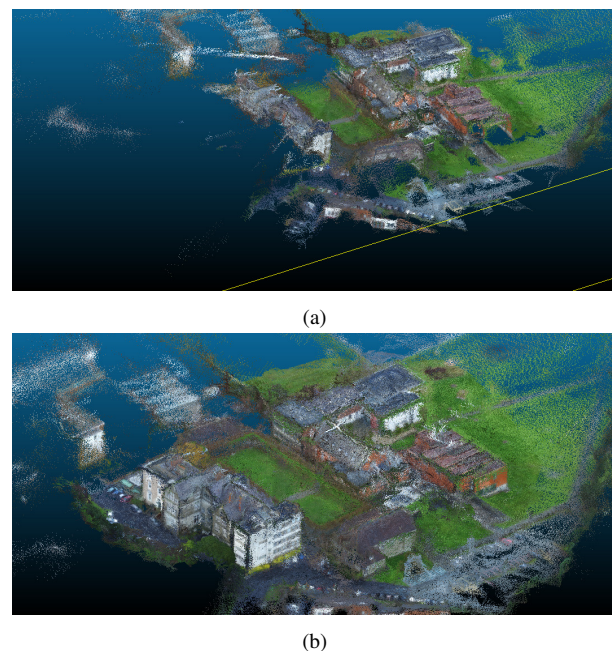


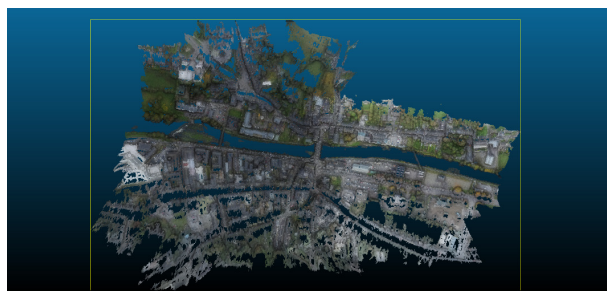
Figure 10. Richview Oblique Reconstruction Difference: (a) Regular. (b) Affine.

	SIFT	A-SIFT	AKAZE	A-AKAZE
# Views	149	149	149	149
# Poses	80	135	56	131
# Intrinsics	1	1	1	1
# Tracks	7,320	89,274	5,001	57,991
# Residuals	35,425	339,443	21,795	220,137
SfM RMSE	0.305	0.391	0.528	0.734
Points After MVE	2,237,159	4,724,107	1,725,561	3,804,753

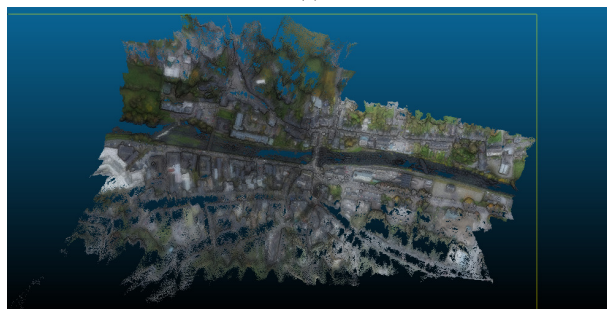
Table 4. Richview Reconstruction Results.

4.2.2 Dataset 2: Nadir Images - Bandon Town The second dataset was captured from a DJI Phantom 3 over Bandon town in county Cork, Ireland. This dataset was tested using the AKAZE descriptor versus the affine invariant AKAZE. As visible in the results in Table 5, there was no increase in image poses found. There was, however, a large increase in the number of tracks from 1.7 million to 5.9 million and residuals from 48,274,000 residuals to 156,000,000 residuals in the final reconstruction. While poses were maintained, the tracking number tripled, thereby boosting the number of correct visible points in the output. The RMSE increased from 0.89 in the original to 0.92 in the affine version. This difference in RMSE would cause a potential difference of 3 cm in the final georectified point cloud model. Additionally, while the reconstruction did not yield a larger number of poses, more information was able to be projected within those poses, thereby demonstrating that the affine invariant AKAZE was able to project a greater level of information into the scene (Figure 13).

4.2.3 Dataset 3: Boyne Bridge This dataset was tested with the SIFT and ASIFT descriptors (Figure 14). The output as shown in Table 6 differed from the other cases. While the number of poses increased by 1, the number of tracks decreased



(a)



(b)

Figure 11. Bandon Town Reconstruction Difference: (a) Regular. (b) Affine.

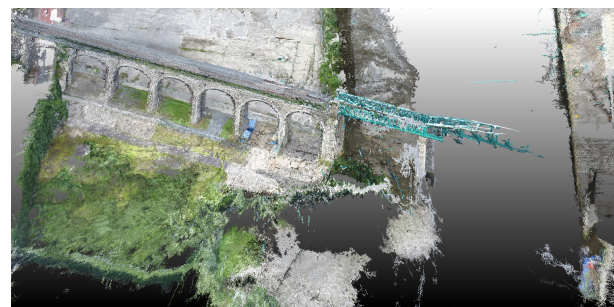
	SIFT	A-SIFT	AKAZE	A-AKAZE
# Views	225	225	255	255
# Poses	225	225	224	225
# Intrinsic	1	1	1	1
# Tracks	314,972	1,352,339	163,191	2,792,571
# Residuals	1,104,957	5,579,303	823,591	7,239,372
SfM RMSE	0.773	0.860	0.938	0.859
Points After MVE	108,918,962	108,476,196	8,702,429	110,954,285

Table 5. Bandon Town Reconstruction Results.

compared to the original SIFT output. This could be due to the similarity of features in the bridge structure. The RMSE of both reconstructions were relatively close with 0.601 for the original to 0.608 for the affine version. Thus, for this particular dataset the affine invariant SIFT descriptor results were similar to the original SIFT descriptor. This example demonstrates that when there are few, large affine transformations between images, applying the affine descriptor is not worthwhile due to the increased amount of processing time required.

5. Discussion

The experimental results demonstrated conclusively that the work of Yu and Morel (Yu and Morel, 2011) can be applied to many other descriptors to obtain affine invariance with a high degree of success. In particular, the binary descriptors such as ORB, BRISK, AKAZE and LATCH gained upwards of 30% tracking in large 3D perspective view changes, although this was accompanied by an average 137% increase in processing time than the original. The advantage of making these binary descriptors affine invariant is that they will then operate with far less memory overhead for descriptor storage than SIFT or SURF, which use many floating point numbers for each detected feature. If certain optimisations are then applied, these



(a)



(b)

Figure 12. Boyne Bridge Reconstruction Difference: (a) Regular. (b) Affine.

	SIFT	A-SIFT	AKAZE	A-AKAZE
# Views	285	285	285	285
# Poses	253	283	261	285
# Intrinsic	1	1	1	1
# Tracks	69,628	290,398	306,482	2,598,295
# Residuals	286,991	1,635,387	875,453	6,323,808
SfM RMSE	0.346	0.521	0.593	0.753
Points After MVE	101,129,717	117,129,717	91,600,741	118,300,067

Table 6. Boyne Bridge Results.

binary descriptors could be useful for operating on larger resolution images that would lead to improved 3D models (Byrne et al., 2017). Further gains could be made through implementation on a GPU. Additionally, (Pang et al., 2012) proposed a method for reducing the affine tilt search space, by identifying the key affine tilts and removing ones that provide limited benefit to get similar results for A-SURF. This type of approach could be undertaken to optimize the other descriptors. This would decrease the total number of simulations required and, thus, reduce the time without negatively impacting the results. If such steps were implemented in frameworks such as OpenMVG (Moulon et al., 2016) the community at large would benefit.

Finally, while usage of the affine descriptor resulted in gains in visual density and improved tracking in the oblique and nadir image datasets, additional information was not generated in the Boyne bridge dataset. Thus, this is not a cure all. With that caveat in mind, the demand for cost-effective photogrammetric based 3D models is high and being able to efficiently generate 3D models of structures and scenes from historic 2D documentation would be particularly beneficial when further in situ documentation is not possible (e.g. after destruction due to an earthquake or armed conflict).

6. Conclusions

Using two datasets (Graffiti and Wall from (Mikolajczyk and Schmid, 2004)), seven OpenCV descriptors were tested. Of those, three had interesting results. The first is BRISK which went from 0.14% inlier matching for image 0 to image 5 to 0.65% in the affine variant. BRISK was also able to maintain good results across both datasets, achieving the best results in the final affine test. ORB also notably improved from 0% to 0.35% with the affine version. Critically, ORB was able to achieve results that would be worthwhile for SLAM, due to its low computational demands (~65 seconds across all tests vs affine BRISK which took ~1013 seconds). LATCH was another binary descriptor that was able to maintain a good tracking number over all the affine tests. In the Graffiti dataset it had an average of ~0.46 and in the Wall dataset it had an average of ~0.63. In terms of speed, LATCH was slower than ORB, but has documented better scale invariance than ORB(?). Affine LATCH would also be a recommendation for a SLAM system. When tested on real world scenes, the affine descriptor resulted in gains in visual density and improved tracking with both oblique and nadir images. In the case studies tested, the affine invariant versions of the descriptors showed a three-fold increase in feature tracking compared to the non-affine invariant versions of those same descriptors. When used with OpenMVG & MVE, the affine invariant improved feature tracking consistently and resulted in 1% to 12.75% more points in the final 3D models than those produced by the regular descriptors.

References

- Abdel-Hakim, A., Farag, A., 2006. Csift: A sift descriptor with color invariant characteristics. *2006 IEEE CVPR'06*, 2, 1978–1983.
- Alcantarilla, P. F., Nuevo, J., Bartoli, A., 2013. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. *Proc. of the British Machine Vision Conference*, BMVA Press.
- Bradski, G., 2000. The OpenCV Library. *Dr. Dobbs's Journal of Software Tools*, 25.
- Byrne, J., Laefer, D., O'Keeffe, E., 2017. Maximizing feature detection in aerial unmanned aerial vehicle datasets. *Journal of Applied Remote Sensing*, 11(2), 025015–1–21.
- Chen, S., Truong-Hong, L., O'Keeffe, E., Laefer, D., Mangina, E., 2018. Outlier detection of point clouds generating from low-cost UAVs for bridge inspection. *Life-Cycle Analysis and Assessment in Civil Engineering, IALCCE 2018*, CRC Press/Balkema, 1969–1975.
- Codreanu, V., 2014. Real-Time Affine-Invariant Feature Extraction: Object Recognition Under Extreme Viewpoint Change. *GPU Tech Conference*.
- Cvišić, I., Česić, J., Marković, I., Petrović, I., 2018. SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *Journal of Field Robotics*, 35(4), 578–595.
- Fuhrmann, S., Langguth, F., Goesele, M., 2014. MVE – A Multi-View Reconstruction Environment. *Eurographics Workshop on Graphics and Cultural Heritage*.
- Ke, Y., Sukthankar, R., 2004. Pca-sift: A more distinctive representation for local image descriptors. 2, II–506.
- Kuai, X., Yang, K., Fu, S., Zheng, R., Yang, G., 2010. Simultaneous localization and mapping (SLAM) for indoor autonomous mobile robot navigation in wireless sensor networks. *2010 ICNSC, IEEE*, 128–132.
- Lodha, S. K., Xiao, Y., 2006. GSIFT: geometric scale invariant feature transform for terrain data. 60660L.
- Lowe, D. G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Intl J. Computer Vision*, 60, 91–110.
- Mikolajczyk, K., Schmid, C., 2004. Affine Invariant Interest Point Detectors. *Int'l J. of Computer Vision*, 60(1), 63–86.
- Moulon, P., Monasse, P., Marlet, R., and others, Pierre Moulon and Pascal Monasse and Renaud Marlet and Others, 2016. OpenMVG. An Open Multiple View Geometry library.
- Neumann, U., You, S., 1999. Natural feature tracking for augmented reality. *IEEE Transactions on Multimedia*, 1(1), 53–64.
- OpenMP ARB, 2021. About Us - OpenMP.
- Pang, Y., Li, W., Yuan, Y., Pan, J., 2012. Fully affine invariant SURF for image matching. *Neurocomputing*, 85, 6–10.
- Porwik, P., Lisowska, A., 2004. The Haar-Wavelet Transform in Digital Image Processing: Its Status and Achievements. *Machine Graphics & Vision*, 13(1/2), 79–98.
- Remondino, F., Spera, M. G., Nocerino, E., Menna, F., Nex, F., 2014. State of the art in high density image matching. *The Photogrammetric Record*, 29(146), 144–166.
- Shibata, A., Fujii, H., Yamashita, A., Asama, H., 2015. Scale-reconstructable Structure from Motion using refraction with a single camera. *ICRA, IEEE*, 5239–5244.
- Sotak, G. E., Boyer, K. L., 1989. The Laplacian-of-Gaussian kernel: a formal analysis and design procedure for fast, accurate convolution and full-frame output. *Computer Vision, Graphics, & Image Processing*.
- Wang, K., Xia, G., Zhu, Q., Yu, Y., Wu, Y., Wang, Y., 2012. The SLAM algorithm of mobile robot with omnidirectional vision based on EKF. *Information and Automation, IEEE*, 13–18.
- Wohlfart, P., Lepetit, V., 2015. Learning Descriptors for Object Recognition and 3D Pose Estimation. <http://arxiv.org/abs/1502.05908>.
- Yu, G., Morel, J.-M., 2011. ASIFT: An Algorithm for Fully Affine Invariant Comparison. <http://dx.doi.org/10.5201/ipol.2011.my-asift>.
- Yu, M., Yang, H., Deng, K., Yuan, K., 2018. Registrating oblique images by integrating affine and scale-invariant features. *International Journal of Remote Sensing*, 39(10), 3386–3405.
- Zheng, M., Wu, C., Chen, D., Meng, Z., 2014. Rotation and affine-invariant SIFT descriptor for matching UAV images with satellite images. *Chinese Guidance, Navigation and Control Conf., IEEE*, 2624–2628.