Performance analysis of interest point detection/matching on shiny and non-textured surfaces

Rick Huizer[1] , **Jan van Gemert**[1] , **Burak Yildiz**[1]

[1]TU Delft

Abstract

3D modeling techniques can be used to automate processes such as damage assessment in aircraft engines. Aircraft engines often have shiny and non-textured surfaces, where these modeling techniques often have poor performance. This paper gives more insight into the performance of interest detection/matching algorithms on shiny and non-textured surfaces as found in aircraft engine borescope inspection videos. These algorithms are often used in 3D modeling techniques. Three interest point detection/matching algorithms are executed on different test videos, and various metrics are calculated for each algorithm. This paper is the first paper that compares both recent and traditional computer vision interest point detection/matching algorithms in these specific settings, and contributes to a better understanding of the usability of feature-based 3D reconstruction techniques. The results show that more recent neural network-based approaches outperform traditional approaches.

1 Introduction

Making measurements in industrial videos is important for, among other things, damage assessment. If a 3D model could be made from an input video and then be used to automatically make measurements, processes like damage assessment would become more reliable and efficient. Even though there are several robust 3D reconstruction methods available, these methods assume input videos to contain plenty of texture resulting in that interest points can easily be matched between consecutive frames. With these matches, a 3D model can be built. Keypoints or interest points in an image are points that define what is interesting / stands out in an image. With a set of interest points of an object, interesting information of this object is captured. However, the surfaces in industrial videos are often shiny and lack texture. 3D reconstruction methods do not perform well in these scenarios, and a partial reason for this is that interest points are difficult to detect on textureless and shiny surfaces. In this paper borescope inspection videos of aircraft engines are the industrial videos that are analyzed. These videos contain shiny and non-textured surfaces and are used for manual damage assessment. If a 3D model could be reconstructed from these videos, manual damage assessment could be assisted or replaced by automatic systems, speeding up both the efficiency and reliability of this process.

As stated in the work of Sperker and Henrich [13], many of the papers that evaluate interest point detectors and descriptors [3,4,8] do so on datasets that consist of images where objects are well textured and have matte or semi-matte surfaces. Since borescope videos contain objects that are shiny and textureless, most existing evaluation work is not that helpful for interest point detection/matching on borescope videos. It is still unclear how these interest point detectors perform on shiny and non-textured surfaces, as found in the borescope videos.

The work of J.Hartmann and J.H.Klüssendorf [4] compares different popular feature descriptors on accuracy and speed on a graph-based Visual SLAM algorithm. Visual SLAM is a 3D modeling technique that is often used when models need to be generated in real-time. The work of Ó. M. Mozos [8] also compares various descriptors on repeatability, invariance, and distinctiveness, in settings that make the results relevant when deciding what detector and descriptor to use in Visual SLAM. The research of S. Gauglitz [3] also provides a quantitative evaluation of detector-descriptor-based visual camera tracking. These papers all agree that SIFT [6] performs better or equivalent compared to other detectors and descriptors in these scenarios.

SuperGlue [11] is a neural network based algorithm to perform interest point matching. The SuperGlue paper evaluates the algorithm on datasets that also contain environments that are not textureless and shiny. This also applies to many other proposed neural network based algorithms such as LoFTR [14], the work by Ignacio Rocco, Relja Arandjelovic, and Josef Sivic [10] and DRC-Net [5]. It is therefore still unclear how these algorithms perform when doing interest point matching on shiny and non-textured surfaces as found in borescope inspection videos.

The related work shows us that there is a question that still needs to be answered:

- *What interest point detection/matching algorithm performs best on shiny and non-textured surfaces as found in borescope inspection videos?*

This research will focus on more recent computer vision ap-

proaches, as more traditional descriptor based interest point matching algorithms often struggle on textureless surfaces.

The paper contributes to a better understanding of the performance of interest point detection/matching algorithms in environments where there are a lot of shiny and textureless surfaces. The paper shows that more recent neural network-based algorithms outperform more traditional computer vision approaches. The best performing algorithms are algorithms that do not just consider the local context of an image but instead also consider the global context, such as LoFTR [14].

In section 2, more general information about the environment and experiments is discussed. A detailed explanation of the conducted experiments is given in section 3, followed by section 4 where the results of these experiments are reported. Section 5 reflects on the ethical aspects and reproducibility of the research, after which section 6 discusses and analyzes the reported results, and gives recommendations for future work. The paper is concluded in section 7.

2 Methodology

To determine what interest point detection / matching algorithm performs best on the shiny / non-textured videos as found in the borescope inspection videos of aircraft engines, each algorithm is executed using the same approach on various videos. After execution, various metrics are computed and used to compare the performance of the different algorithms. This section will first explain what algorithms are compared, and on which videos. After that, the approach to obtain metrics of the different algorithms is explained, along with a motivation for this approach.

2.1 Interest point matching algorithms

This research will focus on more recent computer vision approaches, whilst also using a more traditional algorithm to be able to compare the performance of traditional approaches with the performance of recent approaches. More recent computer vision approaches often utilize neural networks to perform matching of interest points. These neural networks are trained on datasets, whilst more traditional approaches often rely on calculating interest point descriptors to compute matches. Two more recent interest point matching algorithms that will be compared are SuperGlue [11] and LoFTR [14]. The reason that SuperGlue is chosen is because it still has state of the art performance when compared to many other neural network based approaches, and can therefore be used to get a good general understanding of the performance of neural network based approaches. LoFTR is chosen as it is able to detect many high-quality matches especially in regions with low texture, which is very useful for the specific environment found in borescope videos.

The SuperGlue algorithm consists of two parts: interest point detection and interest point matching. The codebase linked in [11] uses SuperPoint [1] as an interest point detector. These detected interest points are then used as input for the interest point matching part of SuperGlue.

The LoFTR algorithm consists of multiple components, including a local interest point detector and a interest point matching component. After the first component detects local interest points using a neural network, the position of the detected points is also taken into account. By including this position, LoFTR can take the *global context* of the image into account when performing the matching of keypoints. It can therefore detect high-quality matches even in regions with low texture. Feature detectors usually struggle to produce repeatable interest points in these low texture regions.

SIFT [6] is used to represent more traditional computer vision approaches. SIFT is chosen as it is still considered a good standard as explained in section 1. SIFT uses various techniques to create feature descriptors that are invariant to scale, rotation and illumination. To calculate interest point matches using SIFT, the feature descriptors in two images are computed, and then descriptors are matched based on a distance metric such as the euclidean distance. Feature descriptors with a low euclidean distance between them are likely to describe the same keypoint in both images.

2.2 Test Videos

The algorithms mentioned in 2.1 are all evaluated on borescope inspection videos of aircraft engines. These videos have many non-textured and shiny surfaces. The camera used to record the videos is inserted into the engine, and then kept static. The rotor blades of the engine are then slowly rotated, such that each rotor passes the static borescope camera. The other parts of the engine do not move around, and therefore the scenes contain both static and moving parts. The videos were provided by Aiir [1], a company that improves aviation borescope inspection videos using artificial intelligence. Frames of example borescope videos can be found in figure 1. The reason these three videos were chosen is because they all have a different degree of texture, varying from very little to a noticeable amount. It is likely that any other video would have a degree of texture that would fall within the range found in these three videos. These videos therefore form a representative video set.

2.3 Manual quantitative evaluation

It is important that an interest point matching algorithm matches interest points that lie on the blades, as the blades are the moving parts of the scene. To be able to fully reconstruct a 3D model, all the blades would have to be modelled. The static background is therefore not that relevant, as it only shows one small part of the engine. Therefore, interest point matches are manually classified into three different categories, as found in figure 2.

As the videos contain many similiar scenes, instead of assessing the whole video only a small fraction of the videos was assessed. This smaller video starts when a new blade enters the frame, and ends when that particular blade is no longer fully visible in the frame. This blade is considered the 'target' blade. The smaller videos last about one to two seconds on average, depending on the video. The three different categories are as follows:

- Correct matches: matches of two keypoints that lie on

[1] Aiir Innovations, see https://www.aiir.nl/

(a) A frame of one of the example borescope inspection videos. The surface heavily reflects the light of the borescope camera.

(b) A frame taken out of another example borescope inspection video. It shows a textureless surface of the blades.

(c) A frame taken out of another borescope example video. This video contains more texture than the other two videos.

Figure 1: Example frames taken out of borescope inspection videos, where the textureless and shiny surfaces can be seen.

the target rotor blade, and correctly match the same keypoint together.

- Irrelevant matches: matches of two static keypoints that are not that relevant for 3D reconstruction. If this would not be taken into account, it could be that an algorithm will produce many correct matches and therefore appear to perform well. If all these matches are from static parts, 3D reconstruction would not yield useful models for damage assessment in example. Matches are also classified as irrelevant when a match connects two points on a different blade than the blade we are tracking in the video.
- Incorrect matches: matches of two keypoints that lie on the target rotor blade, but wrongfully match two keypoints together.

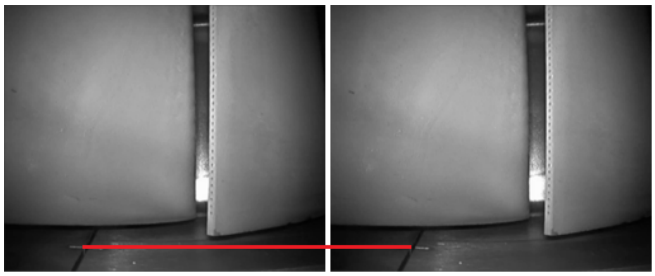
After manually classifying the matches, different metrics are computed to be able to quantitatively evaluate the different algorithms. The metrics are as follows:

- The number of matches detected per frame pair averaged over all frames.
- The percentage of irrelevant matches.
- The percentage of correct matches.
- The percentage of incorrect matches.

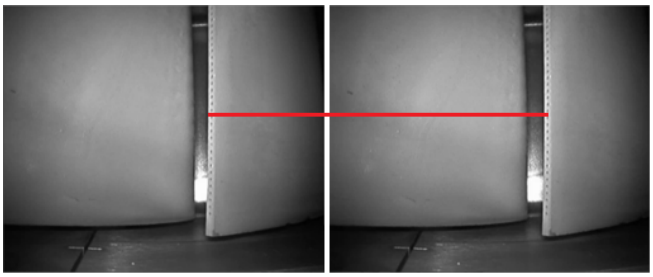
These metrics give a good indication of the performance of the algorithms, as they highlight different relevant aspects. Whilst it is important that the algorithms have an acceptable number of correct matches, 100% correct matches would not immediately imply that the algorithm is useful for 3D reconstruction. It could be that there was only 1 detected match in the image, which would not result in a good 3D model if used as input. A similar argument holds if we would only take into account the number of detected matches.

As can be seen in figure 1, some of the videos contain a camera overlay which can influence the performance of the algorithms. Therefore some of the videos were cropped, an example can be seen in figure 3.

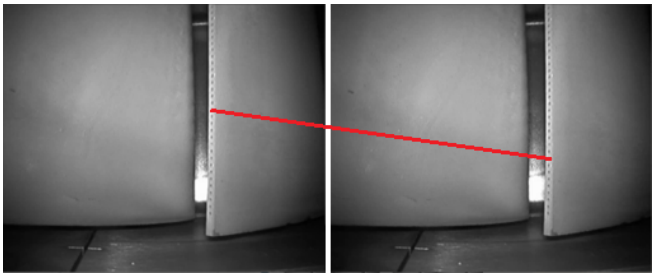
Due to limited time, it was not possible to individually assess all of the matches given by the algorithms. Therefore, for manual assessment, only the best hundred matches were considered each frame pair if the total number of matches was more than one hundred. All the tested algorithms define a confidence score for each match, which was used to sort the matches. The higher the confidence, the more likely the match is correct according to the algorithm. The best one hundred matches were chosen as this would give a decent impression of the performance of the algorithm. If it for example turns out that the top hundred matches already have a poor metric score, then it is likely that all the matches that have



(a) An example of an irrelevant match. The match is part of the static background, and therefore not useful for 3D reconstruction algorithms.



(b) An example of a correct match. The match correctly links two keypoints that are part of the moving parts of the target blade and is therefore useful for 3D reconstruction algorithms.



(c) An example of an incorrect match. The match does link two keypoints that lie on the target blade but wrongfully matches them together. This would have a negative impact on the performance of 3D reconstruction algorithms.

Figure 2: Two frames taken out of two borescope inspection videos, where the textureless and shiny surfaces can be seen.

lower confidence scores will not result in better metrics. Alternatively, if the best hundred matches turn out to have good metric scores, then we can already make some preliminary conclusions about the performance of the algorithm.

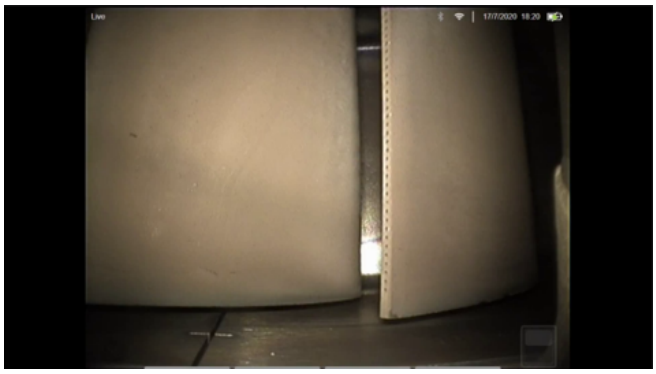
2.4 Automated quantitative evaluation

As only the best one hundred matches are assessed manually, a better understanding of the performance of all the matches is useful. The faster RANSAC [2] approach is used next to manual assessment to automatically assess the matches. RANSAC can be used in combination with calculating a geometric transformation matrix. Two consecutive frames of the videos are related to each other with an affine transformation. There are various approaches to calculate the geometric transformation corresponding to a set of matches, such as calculating the homography, essential or fundamental matrix. The homography matrix can be used when scenes are planar, as that guarantees that there is a projective transformation between two consecutive frames. As the scenes in the borescope videos do not contain just planar scenes, this approach is not suitable. The essential matrix can be used when a calibrated camera is present, but the calibration settings of the cameras used in the test videos is not known. The fundamental matrix approach can be used with non-planar scenes, and is therefore the most suited approach for these experiments.

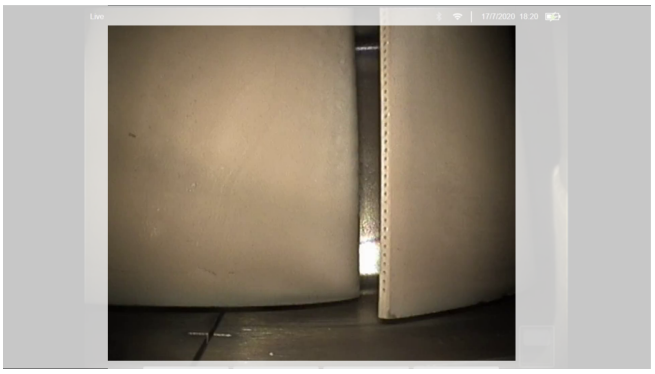
To be able to automatically classify the matches, the fundamental matrix is calculated between two consecutive frames. The fundamental matrix is a 3 x 3 matrix. After multiplying this matrix with the homogeneous coordinates of a point in the first image, the result describes a line on which the corresponding point on the other image must lie. RANSAC uses this matrix as follows:

1. RANSAC will select 4 random matches out of the output of the algorithm.
2. It will calculate the fundamental matrix with these 4 randomly selected matches.
3. The calculated fundamental matrix is then applied to all the remaining matches, and it checks whether the second points of the matches lie close to or on the line found by multiplying the matrix with the first point of the match.
4. A score is calculated to see how many matches were correct with this particular fundamental matrix, and this score is compared to scores of previous iterations. If the score is better, we save this fundamental matrix.
5. Step 1 is repeated until a certain number of iterations is reached.

If there are many iterations, it is likely that the saved fundamental matrix describes the transformation of the movement of the blades between the two consecutive frames. Such a fundamental matrix will often classify matches between the moving blades as correct. Matches that are incorrect or matches that match two static points together do not adhere to the relation defined by the fundamental matrix and are therefore classified as incorrect. An example of this can be found in figure 4.



(a) The video frame before cropping.



(b) The frame after cropping, where white parts represent cropped parts of the image.

Figure 3: An example of a video where the overlay is cropped away.

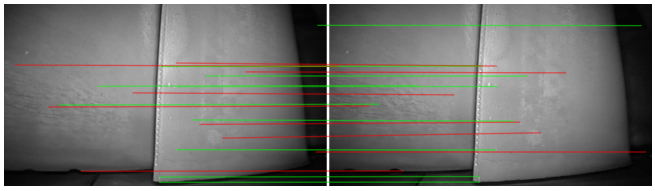


Figure 4: An example of the filtering performed using RANSAC, green lines correspond to correct matches, whilst the red lines are classified as incorrect. Not all lines are drawn to prevent cluttering.

As can be seen in figure 4, there is still a green point that does not lie on the moving blades. A reason why this might occur is that the blades cast a shadow on the static background, which also moves as the blades rotate. RANSAC therefore does not classify these matches as irrelevant, as they are correct according to the fundamental matrix. Another limitation of the RANSAC approach is that matches that do not belong to the target blade of the smaller videos as explained in section 2.3, are classified as correct matches. Even though the match might be correct, it is not useful to create a 3D model of the target blade.

The metrics per pair of frames that are calculated using RANSAC are:

- The average number of matches detected.
- The percentage of correct matches.
- The percentage of incorrect + irrelevant matches.

2.5 Qualitative evaluation

To get a better understanding of the performance of the algorithms, the output of the models is given as input to the 3D modelling algorithm Structure from Motion (SfM) [12]. This algorithm takes the matched keypoints and generates a 3D model represented by a point cloud. These point clouds give a more concrete image of the performance of the different algorithms.

3 Experiment details

This section will go into more detail on how the results were obtained, what exact parameters were used, and explains detailed decisions made in the evaluation process.

3.1 Algorithm setup details

Each algorithm is evaluated on the videos found in figure 1. The evaluation settings per algorithm are as follows:

SIFT

The OpenCV implementation of SIFT was used, with the default parameter values as specified by OpenCV. After the feature points were extracted, the brute force matching algorithm implementation of OpenCV was used to calculate the matches between the feature points. The feature point descriptors were compared using the euclidean distance metric. The brute force matcher takes the descriptor of one feature of the first image and compares it with all descriptors of the second image, returning the closest one. Instead of using the ratio test as described in [6], a cross-check is used to return the most optimal matches. The cross-check ensures that descriptors can only be matched to one other descriptor of the other image.

When using RANSAC in combination with SIFT, the pre-processed images were resized to a resolution of 1280x720 pixels. This was done to be able to directly compare the results with the outcomes of the other two algorithms. The *findFundamentalMat* implementation of OpenCV was used, which has a built-in RANSAC implementation. The default parameter values for this function as found in the OpenCV implementation were used.

SuperGlue

The SuperGlue paper provides a publicly available implementation[2] with pre-trained weights, which was used to evaluate the algorithm. Some pre-processing steps are required to run the algorithm on the various videos, which can be found on the GitHub page. The '*indoor*' pretrained weights were used, and the video frames were not resized when manually assessing the output. For all other parameters, the default values as found in the implementation were used. For RANSAC, the pre-processed images were also resized to a resolution of

[2]SuperGlue implementation, see https://github.com/magicleap/SuperGluePretrainedNetwork

1280x720 pixels. The OpenCV *findFundamentalMat* implementation was used, with default parameter values.

LoFTR

The LoFTR paper also provides a publicly available implementation[3] with pre-trained weights. As explained in the paper, the width and height of the input images should be a multiple of eight, and therefore after cropping the images, they are resized to the closest resolution that is divisible by eight. The '*indoor_ds*' pretrained weights were used. The same implementation for RANSAC was used to perform automatic assessment of the output as for the other two algorithms, along with the same default weights. The images were also resized to a resolution of 1280x720 pixels for automatic assessment.

3.2 Qualitative evaluation

To be able to run SfM on the output of the different algorithms, some pre-processing steps have to be made. SfM requires the matches to be tracked across more than just two frames. The interest point matching algorithms only match image pairs, and these matches need to be converted from just two frames to multiple frames of the videos. An algorithm was developed to track matches across all frames as follows:

1. Perform the image matching algorithm on the first frame pair of the video. Then give all the found matches a unique number and store them.
2. Move to the next image pair and let the algorithm again compute all the matches. Then for each found match, check if the previous frame pair already had a match at that pixel location. If that is the case, label this match with the same number as it was given in the previous frame. If it is not the case, give it a new number as it is a newly detected match.

By repeating step 2 until the end of the video has been reached, all computed matches can now be linked to eachother across multiple frames, and the output can be used for for example SfM or VSLAM.

LoFTR only looks for matches in the reference image at pixels where the pixel coordinates are divisible by eight. The corresponding points in the other image of the image pair can have arbitrary coordinates. The corresponding points found in the second image of the pair therefore need to be rounded to the nearest multiple of eight. By doing this the found matches can be linked across more than just one image pair.

4 Results

4.1 Quantitative results

The results of the manual experiments can be found in tables 1, 2 and 3. All the results are averages per frame, and if there are more than one hundred average matches per frame only the best hundred matches were assessed (see section 2.3). The results of the automated assessment can be found in tables 4, 5 and 6.

[3]LoFTR implementation, see https://github.com/zju3dv/LoFTR

Algorithm	avg # of matches	% irrelevant matches	% of correct matches	% of incorrect matches
SIFT	19.09	67	30	3.0
SuperGlue	101.60	62	29	9.0
LoFTR	740.70	91	9.0	0.0

Table 1: Results of manual assessment of the matches calculated on the video found in 1a. Percentages are based on the best one hundred matches. The abbreviations used are as follows:

Algorithm	avg # of matches	% irrelevant matches	% of correct matches	% of incorrect matches
SIFT	66.05	45	45	10
SuperGlue	555.89	37	62	1.0
LoFTR	5205.60	51	49	0.0

Table 2: Results of manual assessment of the matches calculated on the video found in 1b. Percentages are based on the best one hundred matches.

Algorithm	avg # of matches	% irrelevant matches	% of correct matches	% of incorrect matches
SIFT	226.43	42	56	2.0
SuperGlue	619.30	58	42	0.0
LoFTR	3123.28	74	26	0.0

Table 3: Results of manual assessment of the matches calculated on the video found in 1c. Percentages are based on the best one hundred matches.

Algorithm	avg # of matches	avg # correct matches	% correct matches	% incorrect matches
SIFT	17.25	15.70	91	9.0
SuperGlue	230.05	148.43	65	35
LoFTR	740.70	735.80	99	1.0

Table 4: Results when automatically assessing the matches of video 1a using RANSAC.

Algorithm	avg # of matches	avg # correct matches	% correct matches	% incorrect matches
SIFT	63.40	51.54	81	19
SuperGlue	555.11	246.83	44	56
LoFTR	5205.60	4880.94	94	6.0

Table 5: Results when automatically assessing the matches of video 1b using RANSAC.

Algorithm	avg # of matches	avg # correct matches	% correct matches	% incorrect matches
SIFT	222.64	222.64	87	13
SuperGlue	587.87	334.96	57	43
LoFTR	4760.75	4083.51	86	14

Table 6: Results when automatically assessing the matches of video 1c using RANSAC.

The results of the manual assessment show that in all tested videos, LoFTR detects significantly more matches per frame

than both SIFT and SuperGlue. The fraction of irrelevant matches is highly dependent on what video is being assessed and is roughly the same for SIFT and SuperGlue. LoFTR has a higher fraction of irrelevant matches in all videos compared to the other algorithms. The number of incorrect matches varies per video for SIFT and SuperGlue, whilst LoFTR consistently has the lowest number of incorrect matches. LoFTR has the highest fraction of irrelevant matches in all the videos.

The results of the automatic assessment also show that LoFTR detects significantly more matches compared to SIFT and SuperGlue. The fraction of correct matches is also better than or comparable to SuperGlue and SIFT. SuperGlue detects more total correct matches than SIFT but has a higher fraction of incorrect matches than both SIFT and LoFTR.

In the results of table 4, in a small number of frame pairs, SIFT and LoFTR detected less than 4 matches. Since RANSAC requires at least 4 matches to be able to calculate the fundamental matrix, RANSAC could not be used. The matches of these frames were therefore classified as incorrect. Another interesting observation is that the video as found in figure 1a has significantly worse manual assessment metrics compared to video 1b, and the automatic assessment also shows that the average number of matches is lower. The manual metrics computed on video 1c show that all algorithms have a lower fraction of incorrect matches.

4.2 Qualitative results

The qualitative results can be found in figure 9. Models were reconstructed by the algorithm of Nonnemaker [9], which uses a combination of SfM and Multi-View Stereo.

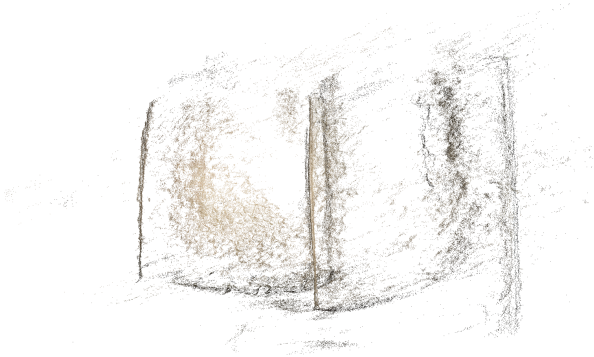


Figure 5: A 3D model reconstructed of video 1b using the input matches of SIFT.

Figure 6: A 3D model reconstructed of video 1b using the input matches of LoFTR.

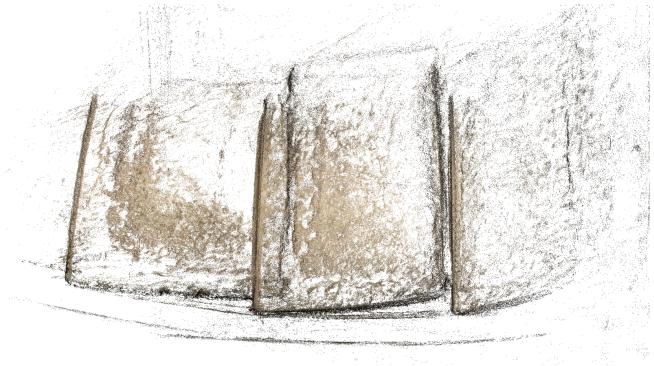


Figure 7: A 3D model reconstructed of video 1b using the input matches of SuperGlue.

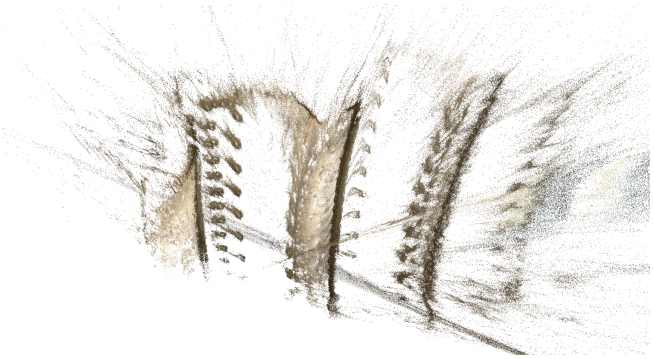


Figure 8: A 3D model reconstructed of video 1c using the input matches of SIFT.

Figure 9: A 3D model reconstructed of video 1c using the input matches of SuperGlue.

The figure shows that the models of SuperGlue are better than that of SIFT and LoFTR. The LoFTR model of the video in figure 1c could not be reconstructed, and the model of the video in figure 1b does not perform according to the expectations found by the theoretical results. The algorithm of [9] did not work properly on LoFTR, for unknown reasons.

5 Responsible Research

The results in this paper show that interest point matching algorithms can be used to reconstruct 3D models of shiny and textureless surfaces. If 3D modeling techniques would actually be used in manual tasks such as damage assessment, it is important that the tested algorithms are not blindly trusted and manual assessment should be used to correct and verify automated assessment. If the automation is done carelessly, it could have a significant impact on the safety of industries such as aviation, as it is not yet clear how these algorithms perform for automated damage assessment. This study only shows that there is potential in the discussed algorithms to be used in environments with shiny and textureless surfaces such as aircraft engines.

Section 3 mentions the settings that were used to obtain the results, together with the details of the environment used. The exact process of how the different results were obtained is explained in sections 2.3 and 2.4. Implementations of the algorithms that were tested are publicly available online, together with pre-trained weights for the neural network approaches. Implementations of any other algorithms that were used such as RANSAC are also available online. The videos used are not publicly available, but similar videos can be found online[4]. Taking all of this into account, the methods and results are fully reproducible for any skilled reader.

6 Discussion

The quantitative results indicate that LoFTR and SuperGlue are interest point matching algorithms that perform better than the more 'traditional' SIFT algorithm on shiny and non-textured surfaces as found in borescope inspection videos. They consistently detect more correct matches than SIFT, which is essential to be able to precisely reconstruct a 3D model of the videos. The qualitative results show that SuperGlue creates better 3D models than LoFTR and SIFT.

One of the main observations is that LoFTR detects significantly more matches than both SuperGlue and SIFT. Both the manual and automatic assessment methods show that LoFTR also has the lowest amount of incorrect matches, where manual assessment even shows that there were no incorrect matches at all. The test video used also has a significant impact on the different metrics. If a video contains more texture, more matches are detected, and also the fraction of correct matches increases, as can be seen in the results of video 1c

The main problem that arises when trying to perform interest point matching on textureless surfaces is that it is hard to find and detect individual interest points, but it is even more difficult to create an associated descriptor vector. More 'traditional' computer vision approaches such as SIFT, use hand-tuned descriptors, whilst more recent approaches such as SuperGlue use learned descriptor representations to perform interest point matching. LoFTR uses a detector-free approach to match interest points, where the *global context* of the image is taken into account. The results also show that learned descriptor representations outperform the hand-tuned descriptors and that the detector-free approach of LoFTR can find many correspondences in regions with low texture, confirming that the results found in [1] and [14] also hold in the context of aircraft engine borescope inspection videos. Even though the qualitative results show that the SfM approach does not work as expected for LoFTR, research done by Markhorst on VSLAM [7] does indicate that LoFTR has potential to be used in 3D reconstruction algorithms. The found results provide a better understanding of what the possibilities are for utilizing 3D reconstructions for applications such as automatic damage assessment of aircraft engines.

The manual evaluation results alone are not sufficient to draw reliable conclusions about the performance of the different algorithms, as it only assesses the best hundred matches per frame. These hundred matches could result in metrics that are not representative of all the matches. To be able to still draw meaningful conclusions, the automated quantitative assessment was done using RANSAC, in addition to qualitative evaluation. These approaches together with the manual assessment give a good general understanding of the performance of the algorithms.

Evaluating the algorithms on additional videos next to the three that are evaluated in this paper would be good to get a better understanding of the performance of the algorithms. Due to limited time available, only three videos could be evaluated. These videos were chosen as they form a representative set of videos together. The degree of texture varies across the videos from very little to a noticeable amount, and any new video that would be evaluated would likely have a degree of texture that would fit within these bounds.

In all the experiments, the parameters of the different algorithms were set to their default values. It is still unclear what the influence of different parameter settings is and what the optimal parameters are. Further research is needed to establish a good understanding of the effect of the various param-

[4] https://www.rvi-ltd.com/borescope.html

eters. Additionally, future research might consider evaluating more algorithms on borescope inspection videos to get a better understanding of what the best algorithm is for matching feature points on shiny and textureless surfaces as found in borescope inspection videos.

7 Conclusion

Automating routine inspections such as damage assessment in aircraft engine borescope inspection videos can lead to more efficiency and reliability compared to manual inspections. Algorithms that reconstruct 3D models of the aircraft engines could be used to automate these inspections. These algorithms often use matched interest points between different images of the engine to reconstruct a 3D model. As it was still unclear how good interest point detection/matching algorithms work in these specific scenarios where there are many shiny and textureless surfaces, this paper has tried to answer what the best algorithm would be for this task. The results show that more recent computer vision approaches based on deep learning have a better performance compared to more traditional computer vision algorithms. Of these recent approaches, SuperGlue appears to perform best in practice, whilst LoFTR has the best performance based on the found metrics. These results show that even in these specific environments with shiny and textureless surfaces, interest points can still be detected and matched with acceptable performance. This means that 3D modeling algorithms can be able to reconstruct 3D models using different matched keypoints. This paper shows that the use cases for 3D reconstruction techniques can also be extended to areas where surfaces are often shiny and textureless, allowing for potential efficiency and reliability improvements.

References

[1] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description, 2018.

[2] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[3] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. Evaluation of interest point detectors and feature descriptors for visual tracking. *International journal of computer vision*, 94(3):335, 2011.

[4] Jan Hartmann, Jan Helge Klüssendorff, and Erik Maehle. A comparison of feature descriptors for visual slam. In *2013 European Conference on Mobile Robots*, pages 56–61, 2013.

[5] Xinghui Li, Kai Han, Shuda Li, and Victor Prisacariu. Dual-resolution correspondence networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[6] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[7] Thomas Markhorst. Performance analysis of simultaneous localization and mapping to reconstruct aircraft engines in 3d, 2021.

[8] Óscar Martínez Mozos, Arturo Gil, Monica Ballesta, and Oscar Reinoso. Interest point detectors for visual slam. In *Conference of the Spanish Association for Artificial Intelligence*, pages 170–179. Springer, 2007.

[9] Alec Nonnemaker. Evaluating structure-from-motion on shiny and non-textured surfaces in borescope videos, 2021.

[10] Ignacio Rocco, Relja Arandjelovic, and Josef Sivic. Convolutional neural network architecture for geometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6148–6157, 2017.

[11] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning feature matching with graph neural networks. In *CVPR*, 2020.

[12] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] Hans-Christian Sperker and Andreas Henrich. Feature-based object recognition—a case study for car model detection. In *2013 11th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 127–130. IEEE, 2013.

[14] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. *CVPR*, 2021.