Loughborough University

This item was submitted to Loughborough's Research Repository by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

Automatic detection of blurred images in UAV image sets

PLEASE CITE THE PUBLISHED VERSION

http://dx.doi.org/10.1016/j.isprsjprs.2016.09.010

PUBLISHER

© 2016 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier

VERSION

AM (Accepted Manuscript)

PUBLISHER STATEMENT

This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0) licence. Full details of this licence are available at: https://creativecommons.org/licenses/by-nc-nd/4.0/

LICENCE

CC BY-NC-ND 4.0

REPOSITORY RECORD

Sieberth, Till, Rene Wackrow, and Jim Chandler. 2019. “Automatic Detection of Blurred Images in UAV Image Sets”. figshare. https://hdl.handle.net/2134/23026.

Automatic Detection of Blurred Images in UAV Image Sets

Till Sieberth*[a,b], Rene Wackrow[a], Jim H. Chandler[a]
(Till.Sieberth@irm.uzh.ch, R.Wackrow, J.H.Chandler, @lboro.ac.uk)
[a]*School of Civil and Building Engineering, Loughborough University, United Kingdom*
[b]*Now at Institute of Forensic Medicine, University of Zurich, Switzerland*

KEYWORDS: Automation, Blur, Image processing, Photogrammetry, UAV

Abstract

Unmanned aerial vehicles (UAV) have become an interesting and active research topic for photogrammetry. Current research is based on images acquired by an UAV, which have a high ground resolution and good spectral and radiometrical resolution, due to the low flight altitudes combined with a high resolution camera. UAV image flights are also cost effective and have become attractive for many applications including, change detection in small scale areas.

One of the main problems preventing full automation of data processing of UAV imagery is the degradation effect of blur caused by camera movement during image acquisition. This can be caused by the normal flight movement of the UAV as well as strong winds, turbulence or sudden operator inputs. This blur disturbs the visual analysis and interpretation of the data, causes errors and can degrade the accuracy in automatic photogrammetric processing algorithms. The detection and removal of these images is currently achieved manually, which is both time consuming and prone to error, particularly for large image-sets. To increase the quality of data processing an automated process is necessary, which must be both reliable and quick.

This paper describes the development of an automatic filtering process, which is based upon the quantification of blur in an image. Images with known blur are processed digitally to determine a quantifiable measure of image blur. The algorithm is required to process UAV images fast and reliably to relieve the operator from detecting blurred images manually. The newly developed method makes it possible to detect blur caused by linear camera displacement and is based on human detection of blur. Humans detect blurred images best by comparing it to other images in order to establish whether an image is blurred or not. The developed algorithm simulates this procedure by creating an image for comparison using image processing. Creating internally a comparable image makes the method independent of additional images. However, the calculated blur value named SIEDS (saturation image edge difference standard-deviation) on its own does not provide an absolute number to judge if an image is blurred or not. To achieve a reliable judgement of image sharpness the SIEDS value has to be compared to other SIEDS values from the same dataset.

The speed and reliability of the method was tested using a range of different UAV datasets. Two datasets will be presented in this paper to demonstrate the effectiveness of the algorithm. The algorithm proves to be fast and the returned values are optically correct, making the algorithm applicable for UAV datasets. Additionally, a close range dataset was processed to determine whether the method is also useful for close range applications. The results show that the method is also reliable for close range images, which significantly extends the field of application for the algorithm.

1. Introduction

Photogrammetry may often involve the use of hundreds of images, collected during image acquisition. Normally the camera platform is stable and the acquired images are of high visual quality. 'Unmanned Aerial Vehicles' (UAV) are a novel technology, which is increasingly used to collect data for photogrammetric purposes. UAVs rarely provide a stable camera platform and are affected by wind, turbulences and sudden operator inputs, which can result in blurred images in the datasets. The detection and exclusion of blurred images is not yet automated. A human operator is required to manually filter blurred images, which is time consuming, exhausting for the eyes and therefore prone to error. It was found that blur detection is rarely an independent topic. Blur detection is often just a preparatory step and provides additional information for blur correction. Current methods are often slow and require extensive calculations, which make them not applicable for photogrammetric datasets. The method presented in this paper provides an automatic approach to detect blurred images, which is both fast and reliable and is therefore applicable for photogrammetric datasets.

Image blurring is a widely used approach in image processing to smooth an image and make it more appealing from a purely visual perspective (OpenCV Dev Team, 2015). This can be achieved using image filters like a Gaussian filter or a Median filter and provides the viewer with an impression of movement or draws their attention to specific objects (NikSoftware, 2013). Detecting whether an image is blurred or sharp remains a complicated process and has not been completely solved. There are a range of different approaches available to

detect blur and it is important to distinguish between optical blur due to poor focus and motion blur due to movement of either camera or object. Furthermore, it is important to distinguish between blur detection approaches that necessitate additional data, such as another image or additional information from other sensors. Methods that do not use additional data are called 'no-reference blur estimation' (Crete et al., 2007). Many different approaches exist in this field, which deal with the question: 'What is blur and how does it manifest itself in an image?' This question can be answered using two main methods that are used in 'no-reference blur estimation'. The first method detects blur based on edge detection and the second detects blur based on frequency analysis.

1.1 Blur detection based on edge detection

Edge detection is a widely used method to detect blur (Ong et al., 2003; Joshi et al., 2008; Narvekar and Karam, 2009). An edge in an image can be considered as a gradient between neighbouring pixels. Edge detection calculates the gradient between neighbouring pixels. In sharp images this contrast is abrupt between two contrasting colour intensities (Figure 1(a)) and edge detection would return a well-defined result for the edge (Figure 1(d)). With increasing blur the contrast decreases and becomes flatter (Figure 1(b)) and the edge detection result returns a flatter gradient over a larger area (Figure 1(e)). In case of largely blurred edges with very flat gradients (Figure 1(c)) edge detection returns barely visible results, or even invisible results due to this gradient being too flat (Figure 1(f)).

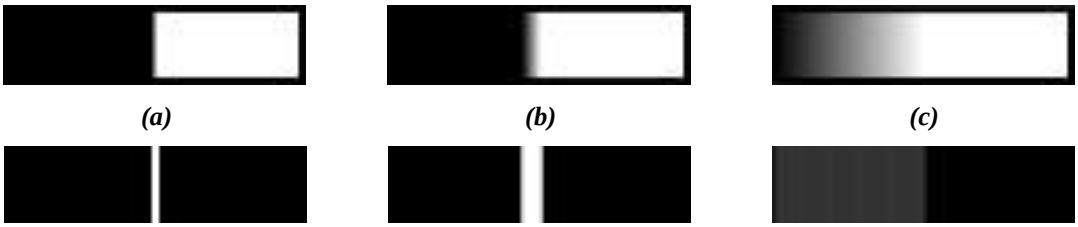


Figure 1. *Edges in an image and the results of edge detection. (a) shows a sharp edge between black and white, which is also represented clearly by the gradient in the edge detection result (d). (b) shows a slightly blurred edge, which results in a larger area with gradients in the edge detection result (e). (c) shows a very blurred edge with a large transition between black and white. The edge detection result shows that the gradient is much smaller but over a much larger area (f).*

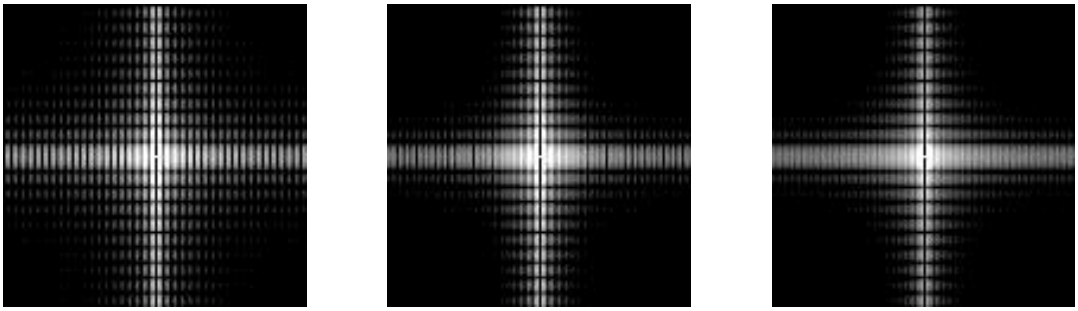
One example of a typical blur detection method is the approach developed by Ong et al. (2003). This calculates the blur value using the average edge-spread value of all the edges in the image and an additional parameter based on subjective ratings, derived from a small group of human subjects. A weakness of the Ong et al. (2003) method is that edge-spread is based on all edges, independent of their orientation. This would be sufficient for optical blur, which is identical in all directions but not for motion blur which depends on the direction of motion. Edges oriented in the direction of blur can influence the result, even if they do not contain any useful information. The method by Narvekar and Karam (2009) calculates the number of edge pixels in sub-images, derived from a small part of the image. After counting the number of edges in the sub-image a decision is made whether or not further processing of the sub-image is required, or if insufficient information is available.

There are several other methods for blur detection based on edges (Jayant et al., 1993; Joshi et al., 2008; Ferzil and Karam, 2009), but they are either similar to the previously presented method, for specific special cases, or are significantly slower in processing images, do return incorrect results or are unreliable.

1.2 Blur detection based on frequency analysis

An alternative approach for blur detection, which does not use edges directly, involves analysing the image in the frequency domain (Rahtu et al., 2012). An image can be represented as a 2D function and described by its frequencies (OpenCV Dev Team, 2014). High frequencies do not appear in blurred images (Liu et al., 2008). The more an image is blurred the less high frequencies are present. Figure 2 shows the results of a Fast Fourier transformation (FFT) derived from the image conveyed in Figure 1(a), 1(b) and 1(c).

The centre is the origin of a coordinate system with lowest frequencies located in the centre and higher frequencies appearing at the boundaries. Figure 2(a) demonstrates that more high frequencies are present than in Figure 2(c), which is based on the blurred example (Figure 1(c)) and this absence of high frequencies can be used to detect blurred images (Rahtu et al., 2012). The frequency approach is an often used approach (Banham and Katsaggelos, 1997; Chen et al., 2011; Chen and Bovik, 2011) but the main weakness of a FFT approach is the computation time to convert the image from the spatial to the frequency domain and back again.



(a) ***(b)*** ***(c)***

Figure 2. *Results of a Fourier transformation on images of Figure 1. (a) is the result of the Fourier transformation of Figure 1(a)which was a sharp edge. (a) shows more high frequencies than the Fourier transformation (b) which is the result for the slightly blurred edge of Figure 1(b). Figure 1(c) results in (c) and has the least high frequencies due to the large blur.*

Also, the analysis of the frequency domain is more complicated than the spatial domain, due to the radial structure of a frequency domain image. This requires transformations from Cartesian to radial coordinates and an offset of the origin from the upper left image corner to the image centre. Another problem is the visual interpretation of frequency images; their abstract nature requires a trained interpreter.

1.3 Blur detection based on other methods

There are several different other methods that have been used for blur detection (Raskar et al., 2006; Lelégard et al., 2012). A useful method developed by Crete et al. (2007) is based on the human perception of blur. This recognises that humans find it difficult to perceive differences between blurred and re-blurred images, but find it easy to distinguish between a sharp image that has been blurred. Figure 3illustrates this and shows re-blurred images of Figure 1. The difference between the sharp image Figure 1(a) and the re-blurred image Figure 3(a) can be detected by the human eye. However, when comparing Figure 1(b) and the extremely blurred image Figure 1(c) to their re-blurred images (Figure 3(b) and 3(c)) the small differences are hard to perceive by the human eye. The differences between the blurred and re-blurred image are not as visible, as between the original sharp and a blurred version. These contrasting differences can be used to detect blurred images as demonstrated by Crete et al. (2007).



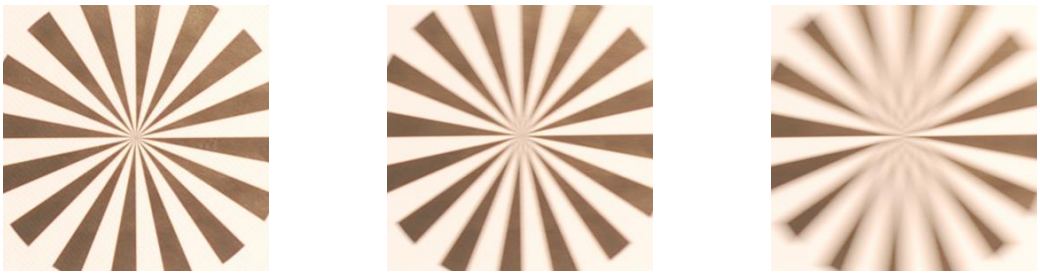
(a) ***(b)*** ***(c)***

Figure 3. *Results of re-blurring the images from Figure 1. (a) is the result after re-blurring Figure 1(a). (b) is the result after re-blurring Figure 1(b). (c) is the result after re-blurring Figure 1(c)*

The algorithm by Crete et al. (2007) involves six well defined steps.

1. Blur the image vertically and separately blur the image horizontally.
2. Compute the absolute variation between pixels both vertically and horizontally from the original.
3. Compute the absolute grey value variation for the rows of the vertically re-blurred image. Compute the absolute grey value variation for the columns of the horizontally re-blurred image.
4. Calculate the difference between the vertical variation images and separately for the horizontal variation images.
5. Summation of all pixel values for both original variation images and the calculated difference images from step three.
6. Normalize the results retrieved in step four.
7. Select either the vertical, or the horizontal value as the blur value (dependent on which one is larger).

As a last step, the authors evaluate the algorithm with a human test to match the computed blur value with a human related perception, the ‘mean opinion score’ (MOS). The algorithm has been made available online from Bao (2009) and was tested with images where the blur was known (Sieberth et al. 2013). The results were disappointing. The extent of blur is particularly visible in the vicinity of the Siemens-Star (Figure 4), which provides an object that has been classically used to illustrate the presence of optical blur (Nasse, 2008). Unfortunately, the Crete et al (2007) algorithm is slow returning results which were also counter intuitive. Figure 4(c) is clearly the most blurred but was classified as less blurred than an image with three times less motion blur (Figure 4(b)). Both, calculation time and reliability, represent serious problems when using huge image datasets, hence the original was believed to be unsuitable for blur detection in UAV image datasets.



(a) ***(b)*** ***(c)***

Figure 4. *Results of Crete et al. (2007) blur detection method. Small values imply only small amounts of blur, larger values mean larger blur. (a) no blur, value based on Crete et al. 0.34. (b) 0.3mm camera displacement, value based on Crete et al. 0.51. (c) 1mm camera displacement, value based on Crete et al. 0.45.*

Most of the reviewed methods are compute intensive, impractical and therefore not suitable for blur detection in UAV datasets. The tests identified in the literature, are very often undertaken on small image sets that have a wide range of backgrounds and a range of different blur sizes. Often this blur is generated mathematically with image filters or exhibits out-of-focus blur and not motion blur. The results are often very subjective as they rely upon human perception. It has been shown that an evaluation of image sharpness and blur is very dependent on the subjective observer (Sieberth et al., 2014(a)). If the image content varies, the evaluation of blur becomes difficult, especially if there is no image for comparison. However, during a literature search it was found that most research focuses on prevention of blur or blur correction. Blur detection is rarely considered an independent topic.

2. Detecting Blurred Images

Deriving a summative value to represent the degree of blur in multiple images is rarely carried out independently, although some measure is used in blur prevention and correction for single images. Without quantification of blur on multiple images, the definition of a threshold value and the automatic exclusion of blurred images from datasets is impossible. Hence, an operator is required to manually identify blurred images and exclude them. Human detection and quantification of blur is dependent on the operator, remains time consuming and is also prone to error. An automatic detection algorithm is therefore required, which can quantify blur in large image datasets and make them comparable to one another.

An algorithm for blur detection was developed in this research. The algorithm is related to the edge sharpness detection algorithm developed by Crete et al. (2007) but utilises concepts based on human perception. As mentioned in Section 1.3 a blurred image can be best detected when it is being compared to another image (Figure 5). If just Figure 5(a) is judged without using any comparison, then is difficult to identify if the image is sharp or blurred. When compared to Figure 5(b), Figure 5(a) appears to be blurred and Figure 5(b) would be judged as sharp image. However, comparing Figure 5(c) to Figure 5(b) reveals that Figure 5(b) is actually blurred too.



(a) ***(b)*** ***(c)***

Figure 5. *Comparing insets of UAV images. To enable a precise judgement the compared image should show the same area or parts of the same area to make judgement of blurred or sharp image possible. (a) most blurred image. (b) less blurred than (a) but more than (c). (c) least blurred image.*

Instead of using different images it is possible to use one input image and blur it synthetically. The aim of the synthetic blur is to generate an image for subsequent numerical comparison. Figure 3 demonstrates that it is possible for a human operator to differentiate between the sharp input image and the synthesised blurred image. However, if the input image is already blurred the differentiation is more difficult or may be incorrect. A stronger synthetic blur has to be applied to enable a human operator to differentiate between the blurred and re-blurred image. The differences between the blurred and the re-blurred image provide the ability for a human to

enable accurate visual blur detection. The approach of using differences between an image and a re-blurred image can be realised in an automatic algorithm and can be used to quantify blur in images.

2.1 The algorithm

The blur detection algorithm aims to detect blurred images in UAV image datasets by using the 'human detection' of blur. A UAV dataset consists mostly of images with a similar texture and colour, typically representing fields, woods but also manmade urban structures. The two requirements of the algorithm should be that it can process the dataset quickly and that it can detect blurred images reliably.
Figure 6 shows the basic steps of the program. These include:

1. Scaling down of the image resolution (Figure 7 (b)).
2. Convert image to hue, saturation, value (HSV) colour space - comprise of a saturation, value, blue (SVB) image (Figure 7(c)). Despite only using the saturation channel in this method the value and blue channel were kept for future comparisons and experiments.
3. Apply low-pass filter (artificial blur) to a copy of the image (Figure 7(d)).
4. Apply high-pass filter (edge detection) on both: low-pass filtered copy and original SVB image (Figure 7(e) and (f)).
5. Calculate difference between both high-pass filtered images (Figure 7(g)).
6. Calculate standard deviation of difference image.

The calculated standard deviation of the difference image is named 'saturation image edge difference standard-deviation' (SIEDS). SIEDS is a single value representing how much an image is blurred. A small SIEDS value represents a small standard deviation in the difference between the original SVB image and the low-pass filtered SVB image, while a large SIEDS value represents a large standard deviation in the difference between the original SVB image and the low-pass filtered SVB image. The larger the SIEDS value the more likely the original input image was initially sharp, while a small SIEDS value indicates that the input image was blurry. This result is similar to that of human perception. The perceived difference between a sharp and a re-blurred image is larger than the difference between a blurred and a re-blurred image.

A SIEDS value can be calculated for each image of an UAV image set and the calculated SIEDS values enables a precise judgement of how much an image is blurred related to other images in the set. However, the absolute calculated values will depend on the processing steps and image content.

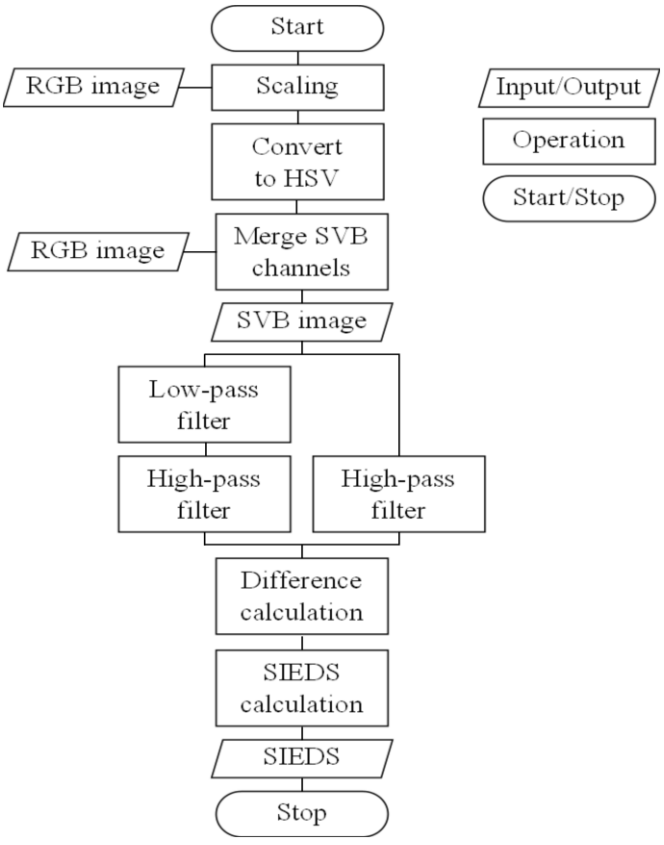


Figure 6. *Flow chart of the developed program.*

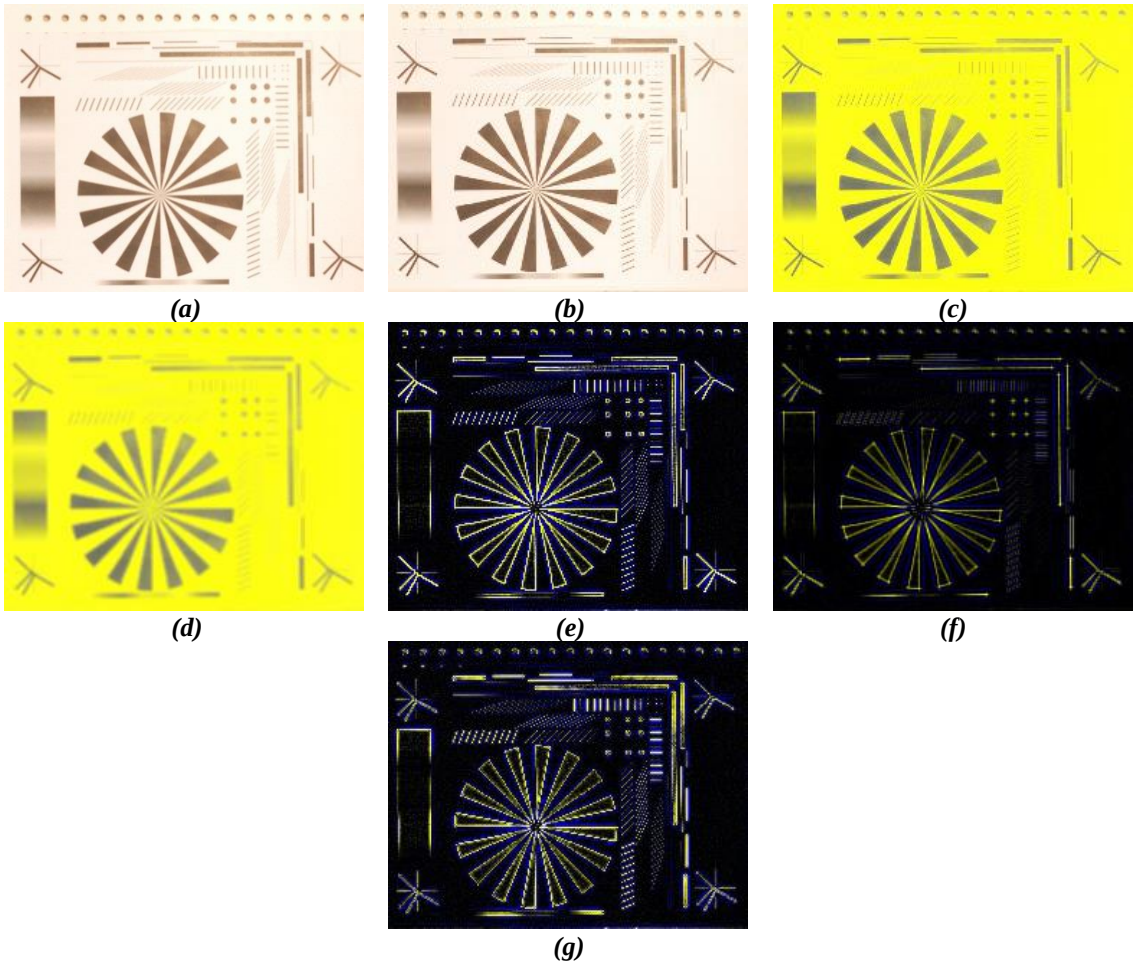


Figure 7. *Steps of the algorithm shown on example image. (a) original image. (b) scaled image (visually identical to (a) due to large image size). (c) SVB image. (d) Low-pass filtered SVB image. (e) High-pass filtered original. (f) High-pass filtered (d). (g) difference image of (e) and (f).*

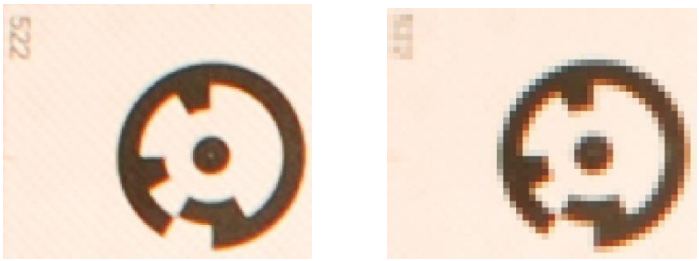
2.2 Detailed explanation of algorithm

2.2.1 Input image scaling

High resolution images contain vast information and inevitably require long processing time. The aim of the algorithm is to calculate the SIEDS value quickly in order to process all the images of a UAV dataset in a reasonable time. One method used to decrease calculation times is to reduce the number of pixels in the image, which is achieved by down scaling the image resolution.

Scaling an image creates several advantages beyond reducing calculation time. It also significantly reduces the required computer memory. In addition, by scaling an image, multiple pixels are combined to just one pixel. When these combined pixels contain the same colour, the newly combined pixel will also have the same colour as all individual pixels. If the individual pixels have different colours, then the newly combined pixel will be a result of the interpolation of the different colours. When this newly combined pixel represents an edge, it will have an impact on determining whether an image is blurred or not. This is useful because homogeneous areas cannot be used for blur detection. By reducing the number of pixels, the number of 'insignificant pixels' representing homogeneous areas is also reduced. However, scaling does not influence edges, which remain important for blur detection.

Furthermore, scaling the image has an advantage of reducing the influence of other effects that appear similar to blur (Figure 8). Spectral mixing and optical errors such as chromatic aberration appear similar to blur and these errors can be reduced by scaling the image. In Figure 8(a) the effect of optical errors are clearly visible, the red and blue contour lines around the target do not exist, but are an effect caused by chromatic aberration. The effect of spectral mixing is also visible, as there is no strict edge between black and white, but a gradient from black via grey to white. By scaling the image it is possible to reduce these effects. The observed gradient in Figure 8(a) occupies 8 pixels and scaling reduces this to just 3 pixels in Figure 8(b).



(a) ***(b)***

Figure 8. *Effect of reducing image resolution. (a) High resolution image with chromatic aberration and spectral mixing on several pixels around the target. (b) Scaled image with reduced number of pixels influenced by chromatic aberration and other optical errors.*

One side effect that needs to be considered during scaling is that the process reduces the effect of motion blur. However, if a camera is displaced during image acquisition then the effect of motion blur maybe obscured by other effects, such as optical errors. When the effect of motion blur is smaller than the effect of optical errors, motion blur becomes undetectable, as it disappears behind optical effects. However, excessive scaling may result in total elimination of motion blur, making detection of blur impossible.

2.2.2 RGB to HSV

The first processing step after scaling an image is the conversion of RGB to the HSV colour space. One of the main differences between optical blur and motion blur is that motion blur is not dependent on the wavelength (colour) of the light, hence colour information is not significant for blur detection. Analysing an RGB image would require analysis of each channel separately, which would take significantly longer calculation times. To eliminate the colour information and reduce the image to just necessary information, the three channel RGB colour image can be converted to a HSV colour space.

The HSV colour space only contains the colour information in the hue channel. Hue does not contain any important information and is not of interest for further processing. It has been observed that increasing image blur results in a reduction of saturation and in value. This observation can be used to detect whether an image is blurred or sharp.

However, for consistency and programming purposes the image matrix should be kept as a three channel image. Furthermore, the additional channels were kept to provide additional information for testing and to enable future calculations of an absolute camera displacement value. To satisfy this requirement, the blue channel is added to the saturation and value channel in order to create the newly composited SVB (saturation, value, blue) image. This technique helps to speed up the calculation process. Furthermore, the image now only contains information that is relevant for blur detection. The fact that the image cannot be converted back to an RGB image is not significant, as a true colour image is not required for the subsequent processing steps.

2.2.3 Re-blurring SVB image

After converting the colour space from RGB to SVB, subsequent image processing steps can be conducted. As determined earlier, the human brain can differentiate easily between sharp and blurred images, but has difficulties in differentiating between a blurred and an even more blurred image (Section 2.3). This ability was identified as a processing step that can be realised in a computer algorithm. To enable a comparison between two images a more blurred image than the original has to be created. This can be done by applying a low-pass filter to a copy of the original SVB image. The copied image is now known to be more blurred than the original input image and can be used by a human operator to determine if and by how much the original image is blurred. Depending on the degree of added blur, it is easier to perceive the additional blur and to determine if the original input image was indeed blurred.

The perception of additional blur largely depends on how much the original image was already blurred and how strongly the image was re-blurred. Adding large additional blur will make the differentiation easier, as the discrepancy between the original and the re-blurred image will be much larger.

2.2.4 Edge detection and discrepancy calculation

Once an image containing additional blur has been created, a comparison can be carried out. To detect blur in an image a human concentrates mostly on the edges, which are represented by a distinct gradient between different grey values. Visual examination of this gradient enables a human to judge whether an image is blurry or not.

This approach can be implemented in a computer algorithm. To detect gradients in an image automatically a high-pass filter can be applied to the original and to the re-blurred image, creating an edge image for both.

For this purpose, a 3x3 Laplace operator is applied to the original and the re-blurred image. To avoid extensive calculations a simple discrepancy image is calculated, which is the difference between the edge image of the original and the re-blurred edge image. In homogeneous areas the discrepancy will be close to zero, while the discrepancy at the edges will be significantly greater than zero. With the discrepancies calculated it is possible to finally calculate the SIEDS (saturation image edge difference standard-deviation) value.

2.2.5 SIEDS calculation

After calculation of the discrepancy image it is possible to carry out the last step to determine a single floating point number, which quantifies the sharpness of the image with the SIEDS (saturation image edge difference standard-deviation) value. To understand the SIEDS value it is important to understand the expected results of the previous processing steps. The gradients in the low-pass filtered image should be lower than the gradients calculated for the original image, additionally, the extrema are 'flattened'. Hence, the standard deviation of the gradients should be smaller for the re-blurred image than for the original. However, neither the standard deviation for the original edge image, nor the re-blurred edge image provides a clear measure about the amount of blur in either image.

The discrepancy image is derived from the original and re-blurred edges. It is expected that the gradients are smaller than in the original image, the amount being dependant on the degree of blur added to the re-blurred image and the amount of blur that existed in the original. If the input image was sharp, then the re-blurred image will have significantly smaller gradients. The discrepancy between the original and re-blurred images will therefore be large. However, if the original image exhibited blur, then the re-blurred image will have similar but smaller gradients and the discrepancies between them will be small.

As a result, the average gradient discrepancies will be smaller or larger depending on the sharpness of the input image. Unfortunately, the average also depends on how many gradients (edges) are available in the image, due to the large number of small values that appear in the homogeneous areas. A rough texture with a large number of edges will create more gradients and increase the average. In comparison, an image with limited texture will return many values close to zero, hence a much smaller average. However, to improve calculations it was decided to use the standard deviation instead of the average grey value. This provides the advantage that the calculation is made independently of how steep the gradients are, and instead uses gradient variation.

3. Results and limits of detection

The saturation image edge difference standard-deviation (SIEDS) value is one single value used to represent the amount of blur for a single image. The value is either large when the image is sharp, or small when the image is blurred. Judgement as to whether an image is blurred or sharp is dependent on all the values, which have been calculated for all images in a dataset. However, critical parameter settings used by the algorithm will also have an influence on the calculated SIEDS values. These settings will be examined in the following sections.

3.1 Influence of image scaling

Modification of the scaling factor influences the calculation time for an image dataset. However, scaling images does not only influence the calculation time, but also the subsequently calculated SIEDS value. To assess the impact of scaling on the calculated SIEDS value, a dataset of 600 images was processed repeatedly, each run containing different extent of scaling. Figure 9 shows the calculated SIEDS values for different camera displacements and differently scaled images.

By scaling the image to a third of the original size the calculated SIEDS value changed more significantly with camera displacement (Figure 9). It was found that the calculated SIEDS value is larger than without scaling. Furthermore, not only is the value increased, but the difference between the largest and smallest SIEDS value is 23 units. This is three times larger than for the unscaled images, which only had a difference of 7 units between the largest and smallest calculated SIEDS value. This large discrepancy enables more precise differentiation between sharp images, images with small camera displacements and images with larger camera displacements. By scaling the image the calculation time decreases further, while the discrepancy between largest and lowest calculated SIEDS value increases to 24 units for a fourth and 26 for one eighth of the original size (Figure 9).

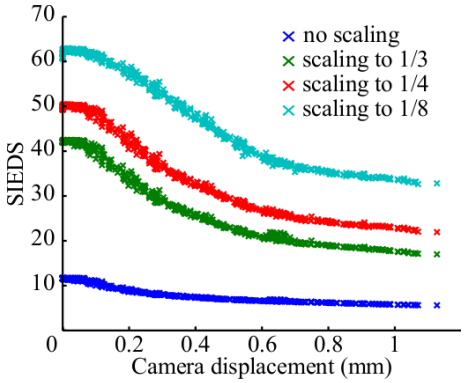


Figure 9. *Influence of scaling on calculation of SIEDS.*

However, the difference between the largest and smallest SIEDS value does not increase significantly and does not provide any advantages. Scaling results in a much faster calculation of SIEDS, but comes with the risk that small structures in the image, which could be used for blur detection, are degraded to a degree that they are not usable.

Based on the quality of the equipment used, the effect of chromatic aberration, spectral mixing, image scale and the contrast of the edges, it was decided that a scaling factor of 1/3 should be used to return optimal results (Section 2.2.1).

3.2 Influence of additional blur

Another setting that influences the calculation of the SIEDS value is the amount of blur that is added to the copy of the image. To visualise the difference in the calculated SIEDS values the images were degraded with different extents of additional blur (Figure 10).

It was found that adding different low-pass filters increased the difference between the SIEDS value of a sharp image and of a blurred image. However, it was also found that the increased difference was not significant. For a 3x3 filter the difference is 23 units, while it is just 24 units for a 9x9 low-pass filter. Even for a larger 81x81 low-pass filter the change did not prove significant. Larger low-pass filters also require longer calculation times and slow the calculation process. The 3x3 filter is therefore judged appropriate.

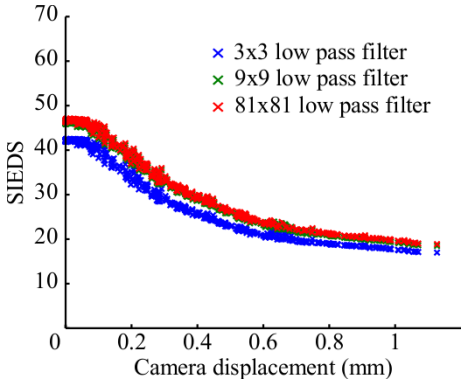


Figure 10. *Influence of low-pass filter size on calculation of SIEDS.*

3.3 Calculated SIEDS

The calculated SIEDS value is derived from the standard deviation of the saturation channel. The use of the value channel of the SVB image or any of the original RGB channels were also tested to see if they could be used instead of saturation (Figure 11).

Figure 11 presents the "SIEDS" values calculated in relation to camera displacements. Beside the SIEDS value based on saturation, "SIEDS" values based on the value and blue channel were calculated also. The results showed that the difference between the largest and smallest SIEDS value is much larger for saturation than for the value and blue channels. The difference for the blue channel is 16 units, while it is 15 units for the value channel. The gradient for the saturation is much larger with 23 units, which enables greater distinction between

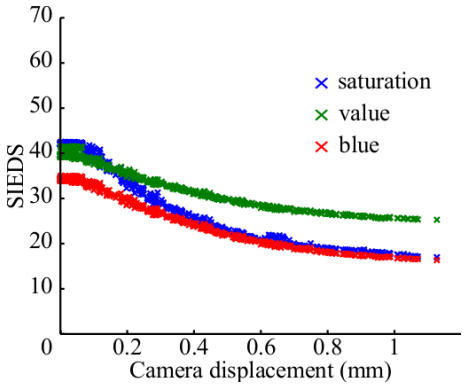


Figure 11. *Different image channels used for SIEDS calculation.*

the sharp and the blurred images. The saturation channel is therefore the most sensitive, which can be explained by considering blur in more detail. When an edge is blurred, the contrast reduces, whilst the colour remains the same. A change in contrast is represented in the saturation of colour, hence saturation is more influenced by blur than value and colour.

By calculating the SIEDS value of one image in isolation it is not possible to judge if an image is blurred or not. Just one single number does not have any significance. To be meaningful it needs to be set in context with other SIEDS values of other, similar images. A similar image here implies a composite spectral response that is averaged across all pixels that are broadly similar. This would be achieved with multiple images of the same terrain type, acquired with the same camera and lens system. Similar terrain therefore suggests areas that show one type of terrain (e.g. forest, urban, agricultural). The requirement that the images show similar terrain is needed for the approach based on edges, which are inherently variable with different types of terrain. However, this is not a problem as UAV flights are normally acquired over one type of terrain.

The SIEDS value is not a random number. It is calculated as a value between zero and a value less than a half the bit depth of the edge difference image, which would be SIEDS=[0, 127) for a standard 8 bit image. The zero value can only be reached if, either every pixel of the image has the same grey value, or all neighbouring pixels have the same gradient, similar to a chessboard pattern. If all pixels have the same grey value then the image would not contain any edges (Figure 12(a)) and so blur detection would become impossible. In case of a homogeneous chessboard pattern, each pixel would have the same gradient, so that the difference between them is zero also and the standard deviation is zero too (Figure 12(b)).

This shows that homogeneous images or repetitive patterns would not be suitable for blur detection using this method. In practice this limitation should not be significant, as UAV images normally show large variations in patterns and colours. Furthermore, these two types of images do not contain any useful information for photogrammetric procedures either, as identification of unique features for coordinate measurement would not be possible either.

The largest practically achievable SIEDS value is achieved by assuming a steep edge caused by one pixel. This SIEDS value would then be 103, which is the maximum SIEDS value that could be achieved theoretically (Figure 13). Clearly, natural images do not produce SIEDS values at these extremes, as they include more diverse gradients. During the development of the program it was found that values between 30 - 60 are more typical.

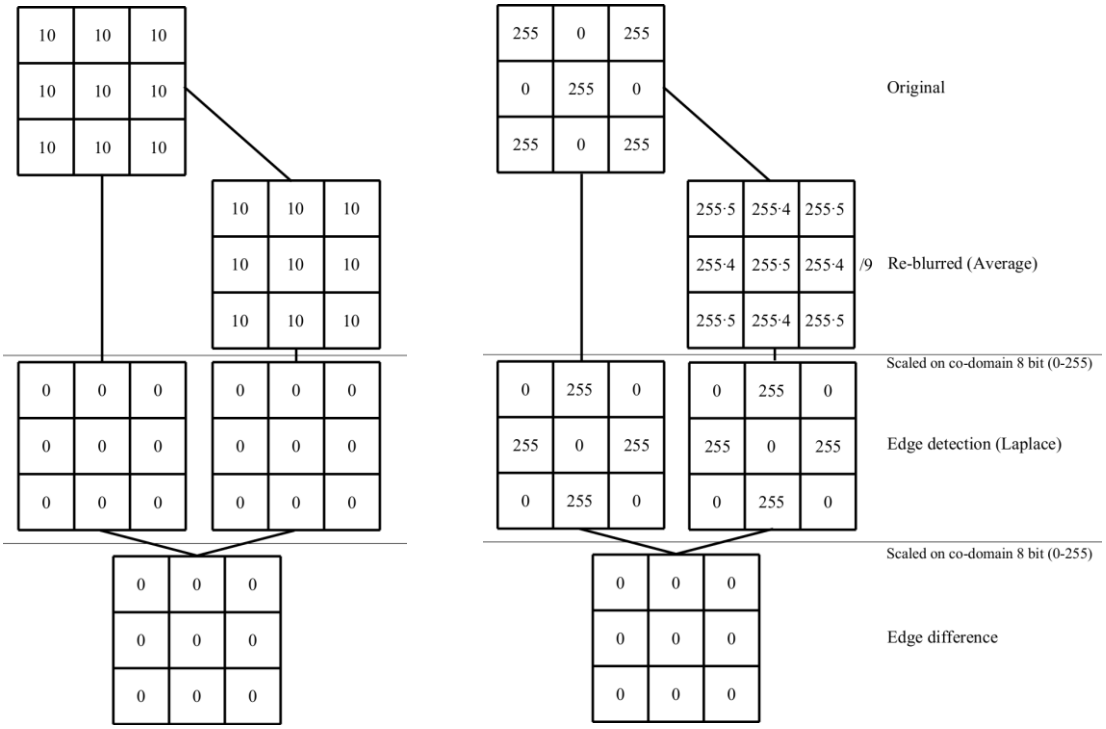


Figure 12. *Minimal achievable SIEDS. (a) homogeneous image would return SIEDS=0. (b) homogeneous pattern would return SIEDS=0.*

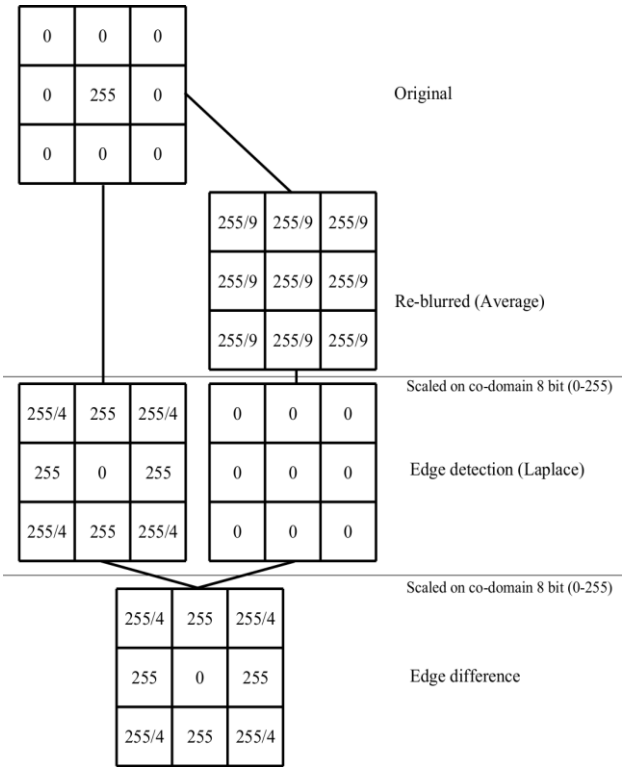


Figure 13. *Calculation for steep edge would return SIEDS=103.*

3.4 Beyond SIEDS

The calculation of SIEDS values for images with known camera displacements reveals that the calculated standard deviation does not change linearly with increasing blur. However, it is obvious that the dependency between SIEDS and camera displacement is continuous and can be described by a function. The function best describing the curve is a special case of the damping function, the over-damped oscillation (Equation 1).

$$b = e^{-\delta\omega}\left(\frac{\delta x_0}{\omega}\sinh(\omega w) + x_0\cosh(\omega w)\right) + n$$

$$\begin{gathered} b \ldots Blur \\ \delta \ldots Damping\ ratio \\ x_0 \ldots Start\ position \\ \omega \ldots Angular\ frequency \\ n \ldots y - offset\ from\ zero \\ \omega \ldots SIEDS, derived\ from\ image \end{gathered} \quad (1)$$

A damping function describes an oscillation, which is damped (Deutsch, 2015). Due to resistance the oscillation is reduced with every wave, which results in zero amplitude after a certain period of time. A special case of the damping function is the over damped oscillation, which means that there is no complete wave before the end of oscillation. A practical example for this function are damped doors. After opening they close but the rate of closure slows before shutting.

It is not surprising that a damping function can be found in images affected by camera displacement. Images are the visualisation of 2D signals, blur damps the signal and in the most extreme case, with an infinite camera displacement an image would appear as an homogeneous coloured 'blotch'. This 'blotch' would have the average colour of the area photographed in the image. The SIEDS value would be large when the image is sharp and high amplitudes are available. With increasing blur, edges in the image would disappear and the SIEDS value would decrease.

The damping function helps to visualise the relationship between camera displacement and SIEDS. SIEDS can be used with the damping function to estimate the displacement of the camera and establish an absolute value that could be compared between different datasets. It was also found that the determination of the other unknown variables, the offset of the y-axis (n), the angular frequency (ω), the start position (x_0) and the damping ratio (δ) are possible, but these are dependent on various factors such as image size, number of edge pixels, average grey value and others, which remains an unsolved problem.

3.5 Modification of the algorithm

There are also other methods to calculate similar measures of blur to SIEDS, which could be considered. For example the standard deviation can be calculated based on the input image, minus the re-blurred image, without applying edge detection. Skipping several steps of proposed algorithm and would potentially be faster, however, the ultimate calculation of a blur value every image pixel would be included and not only edge pixels, which could result in a similar calculation time to the proposed method. The SIEDS value calculated without using edge detection showed again a damping function but the gradient between the sharp images and images with large camera displacements was too small to guarantee correct differentiation between blurred and sharp image.

Another method to calculate SIEDS values includes application of the inverse process to a low-pass filter applied on the input image. It has been investigated if high-pass filtering instead of low-pass filtering could be useful. In this study, the copy of the original image was not treated with a low-pass but a high-pass filter. After applying the high-pass filter the high-pass filtered image was applied on the input image to generate an enhanced image used as comparison. Then both, the input image and the edge enhanced image were processed using a high-pass filter to find the edges and calculate subsequent edge differences. It was found that this procedure produced similar results to using a low-pass filter. However, the calculation of a high-pass filtered image and subsequent enhancing of the input image requires one additional step in the procedure to the calculation of SIEDS. This step takes additional time during the calculation procedure and was considered unnecessary.

4. Application to real world UAV images

Various UAV datasets were processed using the proposed algorithm, each acquired by different UAVs equipped with different cameras and using a range of camera settings. The two datasets chosen for this paper were acquired using a fixed wing and rotary wing platform. The fixed wing UAV was expected to have a certain amount of motion blur in each image due to the forward motion. However, the rotary wing UAV can hover in one position, enabling acquisition of images without forward motion. The rotary wing quad-copter was a kit set that used a *Canon IXUS 500 HS*, 10.1 Megapixel camera (Almond, 2013), while the fixed wing UAV utilised a *SenseFly Swinglet Cam* system (Almond, 2013). Table 1 summarises key characteristics of each dataset.

Table 1. *UAV Datasets used for case application.*

	Fixed wing	**Rotary wing (quad-copter)**
Number images	195	97
Size	12 MegaPixel	10 MegaPixel
Camera	Canon IXUS 220HS	Canon PowerShot S90
Focal length	4 mm	6 mm
ISO	~160	~100
Shutter speed	1/500	1/320
Aperture	f/2.7	f/4
Date	7 October 2012	13 September 2012
Processing time	~1.4 seconds per image	~1.4 seconds per image

Both image sequences were acquired on sunny days, which allowed both fast shutter and film speeds ensuring optimal image quality. Fortunately, the low aperture setting and reduced depth of field had minimal impact because the terrain did not exhibit any significant height differences. Both datasets were acquired of rural salt marsh in Abbotts Hall Farm (AHF) Great Wigborough, Essex, UK, on the North bank of the Salcott Creek, a tributary to the Blackwater Estuary (Figure 14) (Almond, 2013).



(a) ***(b)***

Figure 14. *Examples of real application UAV images. (a) Example image from fixed wing dataset. (b) Example mage from rotary wing dataset.*

4.1 Rotary wing dataset

The first dataset was acquired using the quad-copter and Figure 15 demonstrates that the SIEDS value ranges between 70 and 25, exhibiting a range of sharp and blurred images. A random sample of images was chosen to visual analyse the calculated SIEDS values. The SIEDS values are calculated for the complete image. To enable easy comparison the visual analysis should be conducted on insets showing the same area. The four chosen examples show that overall the dataset appears visually to be of good quality, which can be closer assessed in the insets (Figure 16).

Figure 16(a) is visually the most blurred image. An increasing SIEDS values equates to an improving visual quality. This demonstrates applicability of the calculated SIEDS value and how well the blur detection method works.

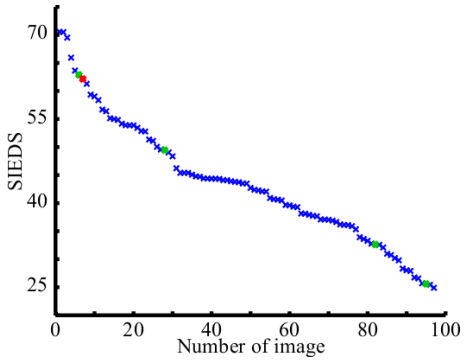


Figure 15. *SIEDS calculated for rotary wing UAV images. The red mark shows the SIEDS value of Figure 14(b). The green marks show the SIEDS values of Figure 16.*

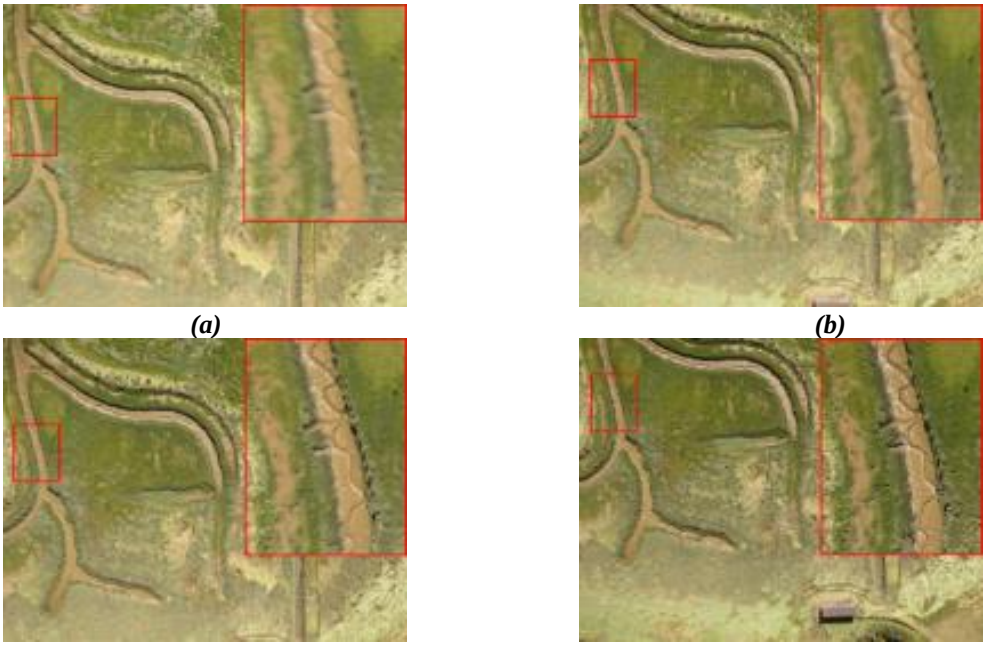


Figure 16. *Example images for calculated SIEDS for rotary wing UAV images. The insets show a more detailed view. (a) SIEDS=25. (b) SIEDS=32. (c) SIEDS=48. (d) SIEDS=63.*

4.2 Fixed wing dataset

The dataset acquired with the fixed wing UAV appeared to exhibit very high image quality also. However, application of the developed blur detection algorithm allowed different levels of sharpness to be detected (Figure 17).

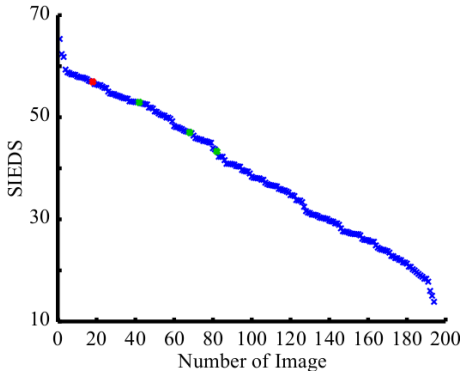
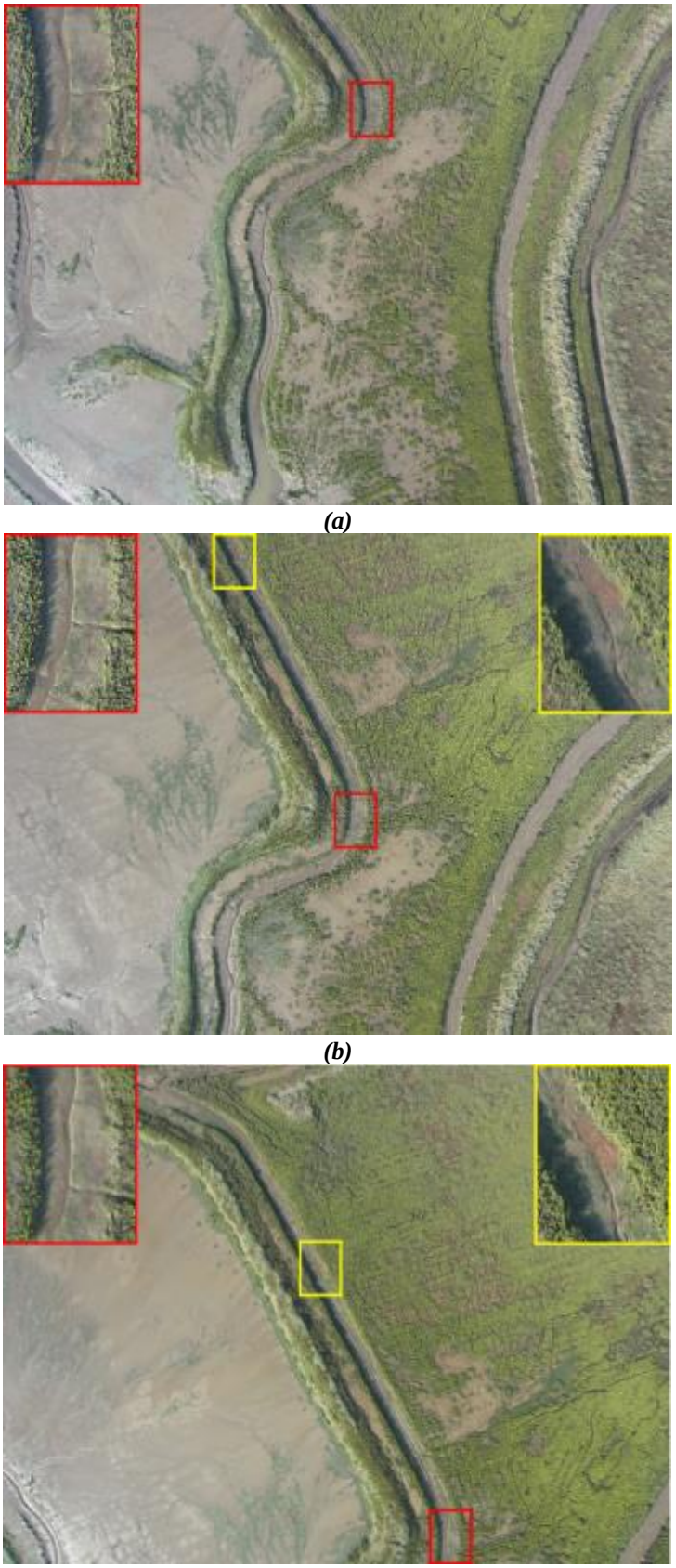


Figure 17. *SIEDS calculated for fixed wing UAV images. The red mark shows the SIEDS value of Figure 14(a). The green marks show the SIEDS values of Figure 18.*



(a)

(b)

(c)

Figure 18. *Example images for calculates SIEDS for fixed wing UAV images. The insets show a more detailed view. The red inset show the same area in all three examples. The yellow inset was only available in two images. (a) SIEDS=42. (b) SIEDS=46. (c) SIEDS=53.*

Figure 17 suggest that all images are reasonably sharp. However, Figure 17 also implies that some images are of extraordinary quality, with SIEDS values above 60, while others are with lower quality, with SIEDS values around 15. The images chosen for comparison have SIEDS values suggesting that they are of similar quality, but the insets reveal differences between the images. The red inset in Figure 18(a) appears to be less sharp than in Figure 18(b), which was expected based on the result determined by the SIEDS value. This confirms that SIEDS represents the quality of an image. However, the same area in Figure 18(c) appears less sharp than Figure 18(b), even when the calculated SIEDS implies that this picture is of better sharpness than the two previous images. This can be explained by the position of the patch chosen for the inset. In Figure 18(b) the inset is towards the centre of the image while it is at the boundary in Figure 18(c). It might be that the camera and lens used to acquire the images introduces distortions at the image boundaries, causing the image to appear slightly out of focus. By choosing another patch (yellow inset), which is in the centre of Figure 18(c) and comparing it to the same area in 18(b), it reveals that Figure 18(c) is indeed sharper.

SIEDS is a value calculated for the entire image, which is problematic when specific parts of the picture are observed. This specific parts of an image used just for visual analysis are referred to as 'insets'. To calculate more precise SIEDS values small areas of the images were used to calculate SIEDS values. This small areas are referred to as area of interest (AOI). By establishing a local SIEDS value for just a small area of interest the SIEDS value changes significantly. The SIEDS value for the AOI in Figure 18(b) (marked red) is the largest with 56, while the AOIs in Figure 18(c) (49) and 18(a) (47) return much smaller values, which are much closer together. This matches with observations when judging the images visually (Table 2).

Table 2. *Comparison of SIEDS values for complete image and image patches.*

Image	SIEDS complete image	SIEDS for red areas of interest
18(a)	42	47
18(b)	46	56
18(c)	53	49

5. Discussion

The visual confirmation of the results calculated by the blur detection program show that the algorithm returns reliable results for UAV imagery. In contrast to frequency domain methods (Liu et al., 2008; Rahtu et al., 2012) the calculation speed is reasonably fast, which makes the method applicable, not only in the office but potentially in the field during image acquisition. This allows the user to acquire new images if necessary and will then avoid the step of blur correction, which remains an unresolved topic in the community.

Currently, a limitation of the SIEDS approach is that different datasets cannot be compared directly to one another due to the varying SIEDS values. It was considered to normalise the SIEDS values within one dataset to enable comparison between different datasets. However, using normalised SIEDS values to make datasets comparable or to determine a fixed "sharpness" threshold is not possible. Datasets may not always contain images exhibiting a sufficiently wide range of motion blur to generate SIEDS values which are comparable in an absolute sense. For example, some datasets might only contain small, acceptable blur for the complete dataset providing SIEDS value in the range : 20-35. Whilst another dataset might only contain a few images with large blur, providing SIEDS values in the range: 15-70. If normalised SIEDS values were generated an overall acceptable dataset would range η=[0, 1] and the same for datasets of unacceptable quality (Equation 2).

$$n_i = \frac{\omega_i - \min(\boldsymbol{\omega})}{\max(\boldsymbol{\omega}) - \min(\boldsymbol{\omega})} \qquad i = 1 \ldots \mathrm{size}(\boldsymbol{\omega}) \tag{2}$$

$$\boldsymbol{\omega} \ldots vector\ of\ all\ SIEDS\ values\ of\ one\ dataset$$
$$\omega_i \ldots SIEDS\ value\ at\ position\ i$$
$$\eta_i \ldots normalised\ SIEDS\ value\ at\ position\ i$$

However, the normalised values would imply that both datasets contain acceptable and unacceptable images. Only the comparison within the dataset gives information of 'how blurry' an image is. To demonstrate this problem more forcibly, both the fixed wing and the rotary wing dataset were normalised independently. Note that these datasets were normalised independently because SIEDS values are dependent on the average brightness of each image. The rotary wing set, which had SIEDS values ranging from 25 to 70, was normalised to the range 0 to 1. The fixed wing dataset with SIEDS values ranging from 14 to 65 was also normalised to the range 0 to 1. The image represented by the insets in Figure 19(a) (fixed wing dataset) had an original SIEDS

value of 50, which was smaller than the SIEDS value for Figure 19(b) (rotary wing dataset) with 58. If these SIEDS values are compared, the rotary wing image (Figure 19b) would be interpreted as less blurred than the fixed wing (Figure 19a), which cannot be confirmed visually. Figure 19(b) is visually clearly more blurred than Figure 19(a). However, after normalising the SIEDS values, both examples would generate an identical normalised value of around 0.65, leading to the assumption that both images are equally blurred, which cannot be confirmed visually (Figure 19).



(a) (b)

Figure 19. *Inter-comparability of UAV blur detection results. Neither the absolute SIEDS value nor the normalised value can be compared effectively. (a) Inset of an image of fixed wing dataset. SIEDS=50. Normalised value=0.65. (b) Inset of an image of rotary wing dataset. SIEDS=58. Normalised value=0.65.*

The appearance of motion blur depends also on the shutter used in the camera. There are two major shutter solutions widely adopted for UAV sensors: the global shutter and the rolling shutter (Red.com, 2015). A global shutter illuminates all pixels of the image sensor at the same time. If the camera is moved during exposure of the sensor, all pixels on the sensor experience exactly the same motion blur. With a rolling shutter however, the pixel lines are exposed one by one. If the camera is moved between the exposures of each line, objects will be displaced compared to the previous pixel line and movement will affect each line differently (Red.com, 2015). Compensation for rolling shutter or distorted images would be possible by weighting the values during calculation of SIEDS. The weighting would be based on the distance to the image centre and could be made dependant on the camera and lens model. An alternative and partial solution involves calculating SIEDS, not for the complete image, but only for areas of interest (AOIs) (Section 4.2). The usage of such AOIs would also be useful if different types of terrain appear in one image. By calculating the SIEDS value for an AOI containing just one type of terrain would make it comparable to other insets of other images with the same terrain.

Furthermore, the use of AOIs to compensate for different terrains or hardware influences also has the advantage that no hardware modification are required (Raskar et al., 2006; Lelégard et al., 2012), which makes the proposed method easy to use. The application to real world images has shown that the method cannot only be applied to UAV datasets, which are suffering motion blur but also to close range datasets.

Table 3. *Comparison of Crete et al. (2007) SIEDS.*

Step	Crete et al. (2007)	SIEDS - Sieberth (2016)
1.		Scaling Image.
2.		Convert to SVB image.
3.	Blur the image vertically and separately horizontally.	Blur the image in both directions simultaneously.
4.	Compute the variation between vertical and horizontal pixels from the original and from the re-blurred images.	Detect edges in both, the original and re-blurred images.
5.	Calculate the difference between the vertical variation images and separately for the horizontal variation images.	Calculate the difference between the edge detection results.
6.	Summation of all pixel values for both original variation images and the calculated difference images from step three.	
7.	Normalize the results retrieved in step seven.	
8.	Select either the vertical, or the	Calculate the standard deviation for all pixels.

horizontal value as the blur value
(dependent on which one is larger).

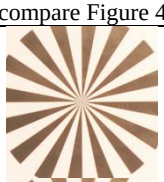
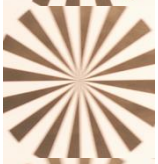
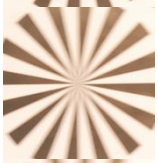
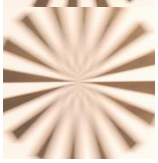
5.1 Comparison with other algorithms

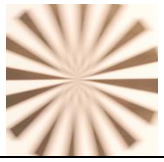
A detailed comparison between Crete et al. (2007) and SIEDS shows that there are several differences between both methods (Table 3). The algorithms involve different processing steps and both the preparatory steps and computations are significantly different. In our implementation, a value for the entire image is computed simultaneously, whilst the Crete et al. (2007) algorithm computes independent values for the rows and columns. The calculation of just one overall value is valuable and allows comparison between images of the same dataset.

The values calculated by the Crete et al. (2007) method, realised in Bao (2009), are different also. Using images with known camera displacements demonstrates that the Crete et al. (2007) method struggles to detect blurred images exhibiting extensive blur (Table 4). Images which experienced camera displacement larger than 0.9 mm were declared less blurred than images which experienced only 0.3 mm of camera displacement. The "Estimated Image Quality" tool in *Agisoft PhotoScan* is able also to differentiate successfully between low quality (blurred) and high quality (sharp) images. However, it is surprising to see quality values larger than 1 because *Agisoft PhotoScan* technical support states that "the quality values range from (0 blurred images) to 1 (sharp images) […]" (Semyonov, 2013). There was no explanation for the AgiSoft Photoscan values larger than one, but this does not matter because it is used here to validate that the algorithm differentiates successfully between sharp and blurred images. However, it does also shows that a normalisation of values is somewhat difficult and does not allow a comparison between different image sets and algorithms.

The algorithm developed in this research calculates a blur value for the entire image, whilst the *Agisoft PhotoScan* value "[…] refers to the area of highest quality. So, for example, images with low DOF [Depth of Field] will have a high quality value[sic], since the focused area is sharp[sic]." (Pasumansky, 2014). This feature could be perceived positively. However, it is suggested that a low DOF image may be misclassified as high quality, would be a disadvantage if a large area of the image is out of focus. The SIEDS algorithm avoids that problem.

Table 4. *Comparison of results between Crete et al. (2007), Agisoft PhotoScan and the SIEDS method using blurred images with known camera displacement. Crete et al. (2007) returns small values for sharp and large values for blurred images. Agisoft PhotoScan returns values in the range from (0 blurred images) to 1 (sharp images). SIEDS returns large values for sharp images and low values for blurred images with an increased numeric range.*

Visual examples, compare Figure 4	Camera displacement	Crete et al. (2007), realised by Bao (2009)	Agisoft PhotoScan	SIEDS
	0 mm	0.3369	1.173	42
	0.08 mm	0.3584	1.060	40
	0.30 mm	0.5073	0.840	29
	0.53 mm	0.5442	0.659	25
	0.91 mm	0.4845	0.514	21



1.03 mm	0.4485	0.491	19

There are also differences in the calculation speed of the algorithms. The Bao (2009) realisation required significantly longer calculation time than SIEDS. In Table 4, the calculation time required by the Bao (2009) was around 15 seconds for each image, which is approximately 10 times longer than the calculation time (1.2 seconds) using SIEDS. The "Estimate Image Quality" tool from *Agisoft PhotoScan* required 3 seconds to process all images. However, it is important to mention that this comparison was based on different programming environments of Matlab in Bao (2009) and C++ in our work. Furthermore, the program developed for SIEDS was realised by an inexperienced programmer, hence it is probable that the program could be more time efficient if programmed by an expert.

5.2 Close range

The results show that the application is able to detect motion blurred images in UAV datasets. To test the algorithm further a close range application involving an image sequence of vegetation was processed (Table 5).

Table 5. *Close range dataset used for case application.*

	Close range dataset
Number images	111
Size	16 MegaPixel
Camera	Nikon D7000
Focal length	85
ISO	1000
Shutter speed	1/20
Aperture	f/5
Processing time	~1.9 seconds per image

The images are unusual because they contain a large amount of blurred background pixels while only the centre of the image, which contains the object, appears in focus (Figure 21). This is caused by the dark light conditions in the laboratory, requiring a wide aperture opening, which causes a narrow depth-of-field. Furthermore, the images do not contain any motion blur because they are taken with a fixed camera on a tripod of a stationary, fixed object.

It was found that the calculated SIEDS values created a similar graph to the one achieved with the UAV datasets. However, the calculated SIEDS values exhibit a narrower range of just 14 units (Figure 20). This can be explained by the large areas which are out of focus, which is responsible for the lower SIEDS value.

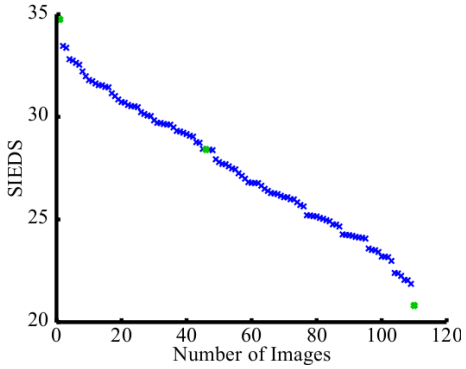
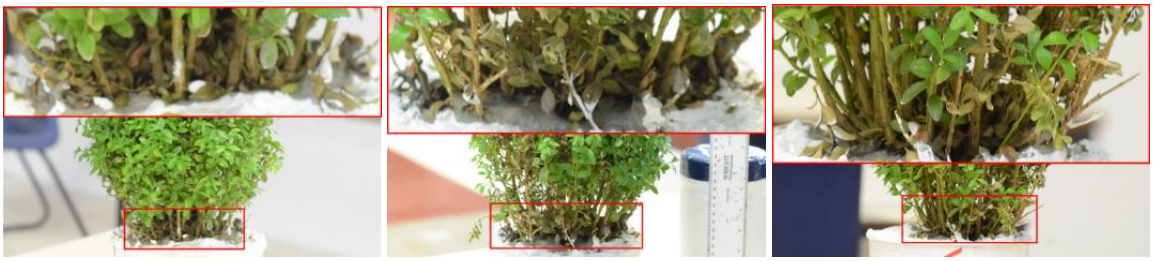


Figure 20. *SIEDS calculated for close range images. The green marks show the SIEDS values of Figure 21.*

The images of Figure 21 contain similar large areas of out-of-focus blur. In the overview in Figure 21, the vegetation appears in focus, however, the insets reveals that the branches of the bush are not of the same level of sharpness. In Figure 21(a) the branches are less sharp than in Figure 21(c). Figure 21(c) appears to be sharpest, which confirms that the calculated SIEDS value represents the level of blur well.



(a) ***(b)*** ***(c)***

Figure 21. *Example images for calculated SIEDS for close range images. The insets show a more detailed view. (a) SIEDS=21. (b) SIEDS=27. (c) SIEDS=35.*

The ability to apply the blur detection algorithm also on close range images extents the algorithm to a much larger field of application. Applying the algorithm to close range images can be useful to distinguish between images of different focusing quality. Distinguishing different focus qualities makes it possible to select only the highest quality images for further processing and prevents errors and problems, which can be cause by blurred images. It is possible that blur causes misdetection and subsequently mismeasurements of photogrammetric targets (Sieberth et al., 2014(a)). The returned SIEDS value could be used to indicate whether or not it is sensible to do image deblurring to ensure reliable target detection and measurement. A calculated SIEDS value for each detected target separately could be used additionally to other information to add a confidence measure describing how reliable the detection of the target is. Feature detection algorithms, such as Scale-Invariant Feature Transform (SIFT), Speed-Up Robust Features (SURF) and similar algorithms, are negatively influenced by blur (Sieberth et al., 2014(b)). The calculation of a SIEDS value could predict and explain the decreased number of feature points and enable the operator to execute a deblurring algorithm or to provide additional images. These are just a few examples for application of the proposed method in the field of photogrammetric applications. However, it is believed that there are many more applications possible.

6. Conclusion and future work

The work described in this paper proves that blur detection in UAV images is possible using the algorithm developed. The fast processing time of the method and the representation of blur using a SIEDS value appears to be sufficient for real UAV image sets. Even high quality images can be assessed, which allows the operator to identify the best quality images fully automatically. Manual filtering would be tedious and require many working hours, which would be prone to error and probably negatively affect the eyes of the operator. Full format aerial and terrestrial images can be assessed with this method also and blurred images automatically excluded.

The calculation of the SIEDS value is dependent on the camera displacement. This can be described by a damping function that could be used to convert SIEDS to a physical value representing the camera displacement. However, the parameters required by the damping function and their dependencies should be researched further with various camera models, different camera displacements and image content. This could perhaps establish a damping model that would enable the calculation of actual physical camera displacement, based on a blurred image. Initial ideas were tested using the value and blue channel but did not return any convincing results to-date.

Although desirable, tests reveal that an absolute SIEDS threshold cannot exist, as absolute values are dependent upon image content, camera and lens. However, it is suggested that SIEDS can be used to filter the lowest quality images. The user can decided then depending on location, geometry, coverage and image content if the image should be excluded. Another approach would be to present the operator a selection of images representing the range of calculated SIEDS values and the operator to establish a threshold for the dataset. It would be desirable to compare further different approaches and evaluate robustness, processing speed and foremost reliability.

References

AGISOFT LLC, (2013), Agisoft PhotoScan User Manual - Professional Edition, Version 1.0.0. Agisoft LLC, 1st edition. http://fieldofviewllc.com/wp-content/uploads/bsk-pdf-manager/27_AGISOFT PHOTOSCAN PRO USER GUIDE.PDF

ALMOND, F., (2013), Using Unmanned Aerial Vehicles and Supervised Classification for Analysis of a Salt-Marsh

Managed Realignment Site. *Loughborough University*

BANHAM, M.R., KATSAGGELOS, A.K., 1997. Digital image restoration. *Signal Processing Magazine*. 14, 24–41.
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=581363

BAO, D.Q., 2009, Image Blur Metric. Internet, accessed: 10/06/2015.
http://www.mathworks.com/matlabcentral/fileexchange/24676-image-blur-metric

CHEN, C., CHEN, W., BLOOM, J. A., 2011. A universal reference-free blurriness measure. *Society of Photo Optical Instrumentation Engineers*, 7867, 1–14.
http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=730916

CHEN, M.-J., BOVIK, A.C., 2011. No-reference image blur assessment using multiscale gradient. *EURASIP Journal on Image and Video Processing*. 2011, 1 – 11.
http://live.ece.utexas.edu/publications/2011/Chen_eurasip_2011.pdf

CRETE, F., DOLMIERE, T., LADRET, P. AND NICOLAS, M., 2007, The Blur Effect: Perception and Estimation with a New No-Reference Perceptual Blur Metric. *Human Vision and Electronic Imaging*, XII, 6492(64920I).
http://proceedings.spiedigitallibrary.org/proceeding.aspx?articleid=1298489

DEUTSCH, J., (2015), Mechanics Ebook. Internet, accessed: 21/04/2015.
http://deutsch.physics.ucsc.edu/6A/book/

FERZLI, R. AND KARAM, L.J., 2009, A No-Reference Objective Image Sharpness Metric Based on the Notion of Just Noticeable Blur (JNB). *Transactions on Image Processing*, 18(4).
http://www.ncbi.nlm.nih.gov/pubmed/19278916

JAYANT, N., JOHNSTON, J. AND SAFRANEK, R., 1993, Signal Compression Based on Models of Human Perception. *Proceedings of the IEEE*, 81(10).
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=241504

JOSHI, N., SZELISKI, R., KRIEGMAN, D.J., 2008. PSF estimation using sharp edge prediction, *Conference on Computer Vision and Pattern Recognition*. pp. 1 – 8.
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4587834

JOSHI, N., KANG, S.B., ZITNICK, C.L. AND SZELISKI, R., 2010, Image Deblurring Using Inertial Measurement Sensors. *ACM Transactions on Graphics*, 29(4).
http://portal.acm.org/citation.cfm?doid=1778765.1778767

LELÉGARD, L., DELAYGUE, E., BRÉDIF, M., VALLET, B., 2012. Detecting and correcting motion blur from images shot with channel-dependent exposure time. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. I(3), 341–346.
http://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/I-3/341/2012/isprsannals-I-3-341-2012.pdf

LIU, R., LI, Z. AND JIA, J., 2008, Image Partial Blur Detection and Classification. In: *Conference on Computer Vision and Pattern Recognition*, IEEE.
http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4587465

NASSE, H.H., (2008), *Wie liest man MTF- Kurven?* Carl Zeiss. accessed: 25/11/2015
http://www.zeiss.com/content/dam/Photography/new/pdf/de/cln_archiv/cln30_de_web_special_mtf_01.pdf

NARVEKAR, N.D. AND KARAM, L.J., 2009, A No-Reference Perceptual Image Sharpness Metric Based on a Cumulative Probability of Blur Detection. In: *Quality of Multimedia Experience*, IEEE.
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5246972

NIK SOFTWARE, 2013, Snapseed. Internet, accessed: 05/07/2013.
http://www.snapseed.com/home/learn/mobile/creative-adjustments/

ONG, E., LIN, W., LU, Z., YANG, X., YAO, S., PAN, F., JIANG, L. AND MOSCHETTI, F., 2003, A No-Reference Quality Metric for Measuring Image Blur. In: *International Symposium on Signal Processing and Its applications*, Volume 1, IEEE.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1224741

OPENCV DEV TEAM, 2014, OpenCV Documentation. Internet, accessed: 09/12/2014.
http://docs.opencv.org/

PASUMANSKY, A., 2014, Estimate Image Quality - Inconsistent Results. Internet Forum, accessed: 21/04/2015.
http://www.agisoft.com/forum/index.php?topic=1924.msg10246#msg10246

RAHTU, E., HEIKKILÄ, J., OJANSIVU, V. AND AHONEN, T., (2012), Local Phase Quantization for Blur-Insensitive Image Analysis. *Image and Vision Computing*, 30(8).
http://linkinghub.elsevier.com/retrieve/pii/S0262885612000510

RASKAR, R., AGRAWAL, A., TUMBLIN, J., 2006. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transactions on Graphics*. 25, 795 – 804.
http://dl.acm.org/citation.cfm?id=1141957
SEMYONOV, D., 2013, Agisoft PhotoScan 1.0.0 pre-release. Internet Forum, accessed: 08/07/2016.
http://www.agisoft.com/forum/index.php?topic=1476.30

SIEBERTH, T., WACKROW, R. AND CHANDLER, J., 2013. Automatic isolation of blurred images from UAV image sequences. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Rostock, Germany, Vol. XL-1, Part W2, pp. 361–366.

SIEBERTH, T., WACKROW, R. AND CHANDLER, J.H., 2014(a), Motion Blur Disturbs - The Influence of Motion- Blurred Images in Photogrammetry. *The Photogrammetric Record*, 29(148).
http://onlinelibrary.wiley.com/doi/10.1111/phor.12082/abstract

SIEBERTH, T., WACKROW, R., AND CHANDLER, J. H., 2014(b). Influence of blur on feature matching and a geometric approach for photogrammetric deblurring. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-3, 321-326
http://www.int-arch-photogramm-remote-sens-spatial-inf-sci.net/XL-3/321/2014/isprsarchives-XL-3-321-2014.html

SIEBERTH, T., 2016. Motion blur in digital images - analys, detection and correction of motion blur in photogrammetry. *Loughborough University*, Loughborough, United Kingdom.
https://dspace.lboro.ac.uk/dspace-jspui/handle/2134/20212

RED.COM, 2015. Global and Rolling Shutter. Internet, accessed: 09/12/2014.
http://www.red.com/learn/red-101/global-rolling-shutter