

Exploiting shadows to infer 3D structures

OMAR GADELMAWLA, University of Twente, The Netherlands

Buildings and vegetation height estimation using remotely sensed images are challenging to achieve. However, effective solutions to this challenge can serve in tackling more complex problems in the remote sensing field that require 3D information about objects in aerial images, which might be costly or inaccessible. Because shadows are a standard metric among many architectural building designs worldwide, shadows can help infer 3D structures as an auxiliary input in Deep Learning (DL) models. This paper proposes a method to combine RGB aerial imagery and height maps extracted from Light Detection And Ranging (LiDAR) sensors to develop an effective algorithm to realistically exaggerate shadow areas in RGB aerial imagery to enhance the learning process of the DL model. The results suggest that the proposed method is an effective solution for the problem, given the evaluation metrics specified in the paper.

Additional Key Words and Phrases: object shadow; remote sensing; deep learning; object height estimation; shadow detection; shadow analysis.

1 INTRODUCTION

Aerial photographs are frequently utilized in geographic information systems (GIS) for a variety of purposes, such as planning for catastrophe prevention and recovery [10, 13], scene change detection, and reconstruction [4, 11]. Aside from city-specific 3D model generation [12]. These aerial images are often a poor source for extracting three-dimensional information because of being two-dimensional (2D). However, in such instances, a Digital Surface Model (DSM), which represents the elevation of the tallest surfaces at that point, is the most common type of supplemental 3D information. It is often created using airborne Light Detection And Ranging (LiDAR), or a Structure Of Motion (SfM) methodology [8, 9]. Height estimation is used in 3D modeling to extract low-cost 3D models. However, calculating the height of objects is a challenging problem that may be solved using techniques, including LiDAR sensors and task-focused deep learning (DL) model [6].

1.1 Related work

LiDAR sensor height maps are used to estimate objects' heights accurately. The LiDAR sensor sends laser light through its transmitter, and the time the light takes to be reflected from the objects' surfaces is used to develop a height map of the objects in the scene, as shown in Figure 2 (right). Despite having some technical inconsistencies when it encounters complex reflective, refractive bodies, and noise, LiDARs remain widely used for DSM acquisition. The expensive flight operation is one of the non-technical limitations of using LiDARs to obtain DSMs.

Another approach was to develop and train a DL model to predict the DSM based on the architectural features of the buildings [6]. However, these features vary from one location to another; as a

TS&IT 37, July 8, 2022, Enschede, The Netherlands

© 2022 University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

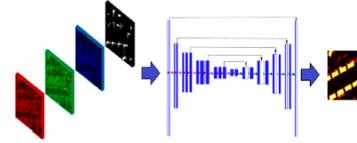


Fig. 1. The shadow map is fed beside the RGB color channel information.

result, it does not perform as well outside of the area data set it was trained for.

In addition to using the design characteristics, the model is further enhanced to include shadow information at the model's input [7]. A shadow map is fed beside the RGB channels as auxiliary information, as shown in Figure 1, to improve the model prediction results.

Moreover, various researches have been conducted to detect shadow areas in satellite effectively aerial imagery [1], in addition to research on how to calculate buildings' heights using their shadow's length [14].

A literature review of the previous related work is conducted to analyse the problem and extract the valuable resources for this research to help understand how shadows in RGB images can be detected and exaggerated realistically. The ideas and methods extracted are further implemented, and their corresponding results are evaluated according to specified metrics.

1.2 Problem statement

In this project, a research will be carried to understand how to detect and exaggerate shadow areas in RGB aerial imagery, in which will provide an auxiliary enhanced shadow map as an input for the model.

1.2.1 Research Questions. How can shadows present in images be realistically exaggerated to facilitate the prediction of height maps when using aerial RGB images and LiDAR map technologies?

RQ1: How can we create synthetic data suitable for training DL models to better learn from shadows in RGB images to predict objects' heights more accurately?

RQ2: How to create a model/algorithm to realistically draw artificial shadows in RGB images?

2 TOOLS AND TECHNICAL ASPECTS

2.1 Data set

The data set used consisted of 100 different aerial RGB images and their corresponding LiDAR height maps in NumPy formatted files.

The LiDAR height maps in the data set are represented as 64 x 64 NumPy arrays, while the RGB images in the data set are represented as 640 x 640 NumPy arrays.

An example of both RGB image and LiDAR map is shown in Figure 2.

2.2 Libraries and tools

NumPy scientific computing library and OpenCV image processing library were the main two libraries used in this project.

Since the LiDAR height maps resolution is ten times less than the resolution of the RGB images, OpenCV interpolation functions were used to upscale the LiDAR maps. This was done because on many occasions information has to be retrieved from the exact pixel location in both the LiDAR map and the RGB image. This has provided a convenient one-to-one mapping between the RGB image and the LiDAR map.

2.3 Assumptions

Two assumptions have been made and considered throughout this project:

- (1) All the images in the data set are taken at approximately the same time; therefore, they have the same shadow direction (Top-left).
- (2) The airborne camera carrying the sensor is perpendicular to the image scene; therefore, the camera observes the total shadow projected by the objects.

2.4 Shadow detection

For the shadow areas in the RGB images to be exaggerated, they should be detected first. In earlier project stages, a method was used to detect a shadow pixel by checking if the blue component in the RGB was the highest among the other components and by comparing the mean value of the RGB color components to a certain threshold. This method was changed later during the progress as it has shown low efficiency in avoiding the detection of false-positive shadows presented in low saturated areas. As a result, the method has changed, and shadow areas were detected by extracting a separate shadow map from the image. This method is explained in detail, besides some examples later in the paper.

2.5 Object detection

In this project, we will depend on LiDAR height maps to detect objects. Objects need to be detected for the following reasons:

- (1) To know which objects are casting a shadow
- (2) To avoid exaggerating shadows on higher structures.

The objects' pixels will be detected by comparing their height in the height map to a certain threshold. A value threshold of 1 meter is chosen to include cars and vegetation beside the buildings in the exaggeration process.

3 METHODOLOGY AND APPROACH

3.1 Trigonometrical exaggeration

The authors in this paper [14] have explained three methods to estimate building heights based on their projected shadow. The methods differ according to the position of the LiDAR sensor with respect to the sunlight direction. The positions mentioned were used to calculate the sun and the sensor azimuth (The angle of elevation of the sun and the sensor) and to calculate the building height accordingly. Since we already know the objects' height from the LiDAR map, a method from the ones proposed can be chosen

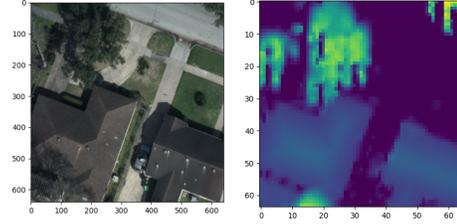


Fig. 2. The original RGB image (left) and its corresponding LiDAR height map (right)

to calculate the shadow length from the object's height. The three conditions of the LiDAR sensor and sun positions are listed below:

- (1) The sensor and the sun azimuth angles are the same (sensor only observes a part of the shadow projected by the building)
- (2) The azimuth angle between the sensor and the sun is greater than 180° (sensor observes all of the shadow projected by the building)
- (3) The azimuth angle between the sun and the sensor is within 0° and -180° . (sensor azimuth influences the shadow detection method)

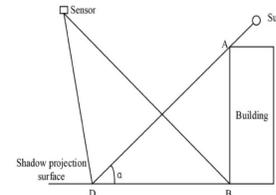


Fig. 3. The azimuth angle between the sun and the sensor is greater than 180° [14]

The second method was chosen based on assumption 2 in the previous section and for the lack of information about the sensor's azimuth.

3.1.1 Calculating the sun azimuth. An equation was derived from Figure 3 to calculate the sun azimuth using the shadow length observed in the RGB image. See the equation below:

$$(\alpha) = \tan^{-1}(AB/BD)$$

Where (α) is the sun azimuth, AB is the building height acquired from the LiDAR height map and BD is the observed shadow length calculated by giving the coordinates of an object and its corresponding shadow manually to the algorithm.

3.1.2 Shadow length calculation. The shadow length calculation has taken place using the following equation which has been derived from Figure 3 as well:

$$BD = AB/\tan(\alpha)$$

The next step is to use the calculated sun azimuth to calculate the length of the artificial shadow to be drawn for each object. As a

result, each pixel in the object will have a corresponding shadow length calculated and an artificial shadow drawn accordingly.

3.2 Progressive exaggeration

This method will be explained by a number of the logical steps taken below:

- (1) The LiDAR map is first resized to match the size of the RGB image
- (2) For each detected pixel above the height threshold, an artificial shadow layer is added to its shadow edge (if and only if there is a shadow projected by the object)

To briefly explain, the algorithm adds an artificial shadow layer progressively at the edges of the object shadow. The logic here was to have a part of the object's shadow exaggerated and re-use the artificially drawn shadow to expand it further using the object's remaining pixels. An example of a building roof progressive exaggeration is shown in Figure 4.

Furthermore, a factor map was extracted from the algorithm's output. The factor map calculates a number between 0-1 for any pixel whose shadow is exaggerated. The factor is calculated by dividing the artificial shadow length by the natural shadow length. This has allowed extracting an auxiliary input for the DL model, so it knows which pixels are exaggerated by which factor, as shown in Figure 5.



Fig. 4. (From the left) 1) RGB original image, 2) image after 5 iterations, 3) image after 20 iterations, 4) output of the RGB image

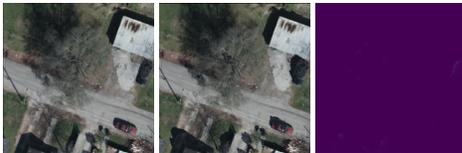


Fig. 5. An example of an RGB image (left), RGB exaggerated image (middle) and its corresponding exaggeration factor map (right)

3.3 Shadow map dilation

This approach is dependent on the morphological operations of erosion and dilation in image processing [2, 3, 5]. Image erosion removes pixels from the object's boundaries to eliminate irrelevant details, while image dilation adds pixels to the boundaries of the objects. In other words, erosion shrinks the object while dilation expands it. Both operations require a kernel and an anchor point to be specified. A kernel matrix is the size of the structuring element; it determines the change of the value of any given pixel by combining it with different amounts of the neighboring pixels. While the anchor point is the position inside the structure element.

The LiDAR map is first resized in this method to match the RGB image dimensions using cubic interpolation. Then the RGB image is converted to gray-scale by getting the dot product of the RGB image matrix and RGB weights matrix with the following values for RGB, respectively [0.2989, 0.5870, 0.1140]. Then the pixels of the gray-scale image with a value less than a value threshold have their RGB values copied to a new image; otherwise, a black RGB value is copied. The threshold used in this method is 60. This has been chosen by experimenting with different values that work for the whole data set. Only a few images have lighter shadows than the others, but as we increase the threshold, the number of false shadow pixels detected increases.

To avoid exaggerating shadows on higher structures, the pixels with a height above the threshold (1m) in the corresponding LiDAR map are also filtered and assigned the color black in the new RGB image. This advantage has been given by having the LiDAR maps corresponding to each RGB image to avoid the false positive building shadow detection problem rising in [1]. The RGB values of the shadow parts in the new image are made slightly darker using a dimming factor. Until this step, we have a clean shadow map, as shown in Figure 6. In some cases, various scattered shadow pixels (false positives) appear in the low saturation RGB images' shadow maps due to detecting dark pixels in roads and vegetation. To deal with this problem, an erosion is performed to blend the scattered pixels with the surroundings, leading to a less noise shadow map, as shown in Figure 7. Sometimes the erosion causes some shadows to disappear, especially at the edge of the trees where the shadow is not solid, but this has not caused significant problems and is still efficient in solving the false shadow scattering issue. This method uses a kernel matrix of size 5 x 5 for the dilation and erosion operations. An anchor point [0,0] is used for the dilation operation in the top-left direction, while the anchor point for the erosion operation is set to default. However, this can be further changed after the inspection conducted in the evaluation section.



Fig. 6. Original RGB image (left) and its extracted shadow map (right)



Fig. 7. Shadow map before erosion (left) and shadow map after erosion (middle), shadow map after dilation (right)

The output of the eroded shadow map is dilated for some iterations, one of them to expand the large shadow areas after being

eroded with the more undersized areas which now have disappeared, and the remaining iterations to exaggerate them further, as shown in Figure 7. The increased darkness in the exaggerated shadowed areas allowed a clear recognition between the exaggerated and natural shadow areas. An example of the final output of the shadow map is shown in Figure 8.



Fig. 8. Original RGB image (left) and its corresponding exaggerated RGB image(right)

4 FLOWCHARTS

Each of the methodologies discussed in the previous section is illustrated using flowcharts.

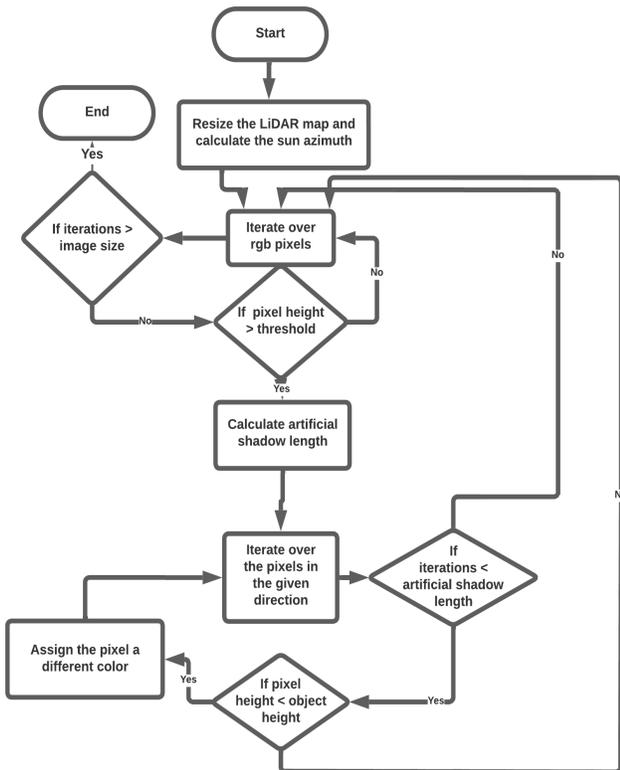


Fig. 9. Flow chart corresponding to the trigonometrical methodology

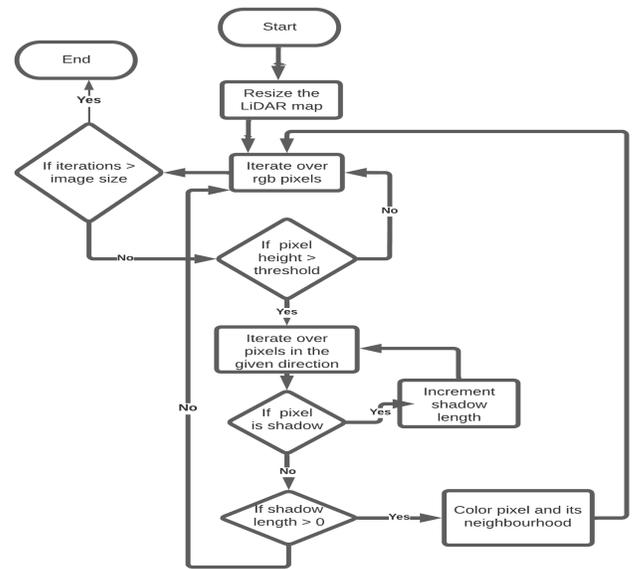


Fig. 10. Flow chart corresponding to the progressive exaggeration methodology

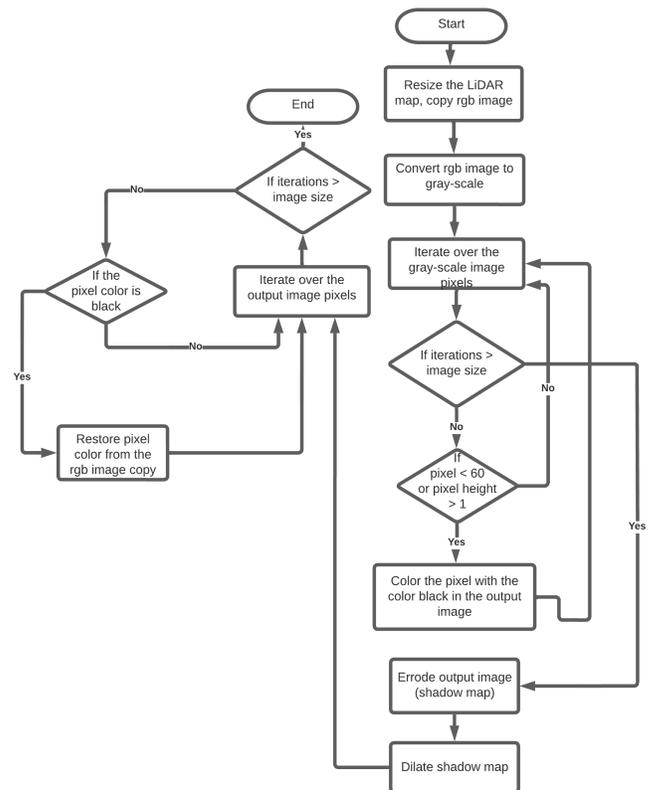


Fig. 11. Flow chart corresponding to the shadow map dilation methodology



Fig. 12. problem 1 corresponding to the trigonometry methodology (Red), problem 1 corresponding to the trigonometry methodology (Yellow), problem 1 corresponding to the trigonometry methodology (Green)

5 EXPERIMENTAL RESULTS

The overall results of each method will be described separately in this section. An analysis will be made on each of them to accept or reject the result's corresponding methodology.

5.1 Trigonometrical methodology

The results of the trigonometry-based method have shown various flaws and inconsistencies as specified below:

- (1) Artificial shadows were drawn for objects which do not cast a shadow.
- (2) Validating the correctness of calculating the sun azimuth was challenging as various images do not adhere to the two assumptions mentioned earlier in the document. However, based on the results' inspection, many objects had their shadow over-exaggerated, meaning that the exaggerated shadow is much longer than the object's height. Therefore, the method's correctness has failed to be validated.
- (3) Some images in the data set have different shadow directions. As a result, a set of coordinates had to be given manually to the algorithm to calculate the sun azimuth for each image which is inconvenient and time-consuming.

The reason for problems 1 and 2 was algorithmic; the algorithm failed to avoid problem 1 and failed to validate the correctness of the sun azimuth calculation as in problem 2. In contrast, problem 3 was caused by the data set inconsistency; the data set contained many images with different shadow directions. The method was used before to estimate buildings' heights using its observed shadow, having a lot more information about the sun's and sensor's azimuth [14]. Since this information was lacking in this project, the method has not performed as expected and, therefore, has been refused from the early stages. Some examples of the inconsistencies are shown in Figure 12.

5.2 Progressive exaggeration

Due to the problems in the trigonometrical approach stated in the previous subsection, this method has been developed and considered these flaws. From the results' inspection, the method has solved problems 1 and 2. Problem 1 was solved by exaggerating objects if and only if they are casting a shadow. In contrast, problem 2 was solved by eliminating the sun azimuth calculation and exaggerating the object according to its shape.

This method has shown exciting results in images with low noise in their corresponding LiDAR height maps and with the same shadow direction given to the algorithm, as shown in Figure 13.



Fig. 13. An example images of how the progressive exaggeration algorithm has performed well in few cases

However, the method has shown unsatisfying results in the cases stated below:

- (1) Self shadows are detected in many images, and as a result, some false shadows were shown on the top of the buildings and some vegetation areas.
- (2) Some objects in the RGB images not placed in the same position in their corresponding LiDAR maps (such as moving cars) have their shadow non-exaggerated, as shown in Figure 17.
- (3) Images with a different shadow direction have their shadow non-exaggerated due to failing to detect a shadow in the specified direction.
- (4) Some objects have contained gaps in their projected shadow, resulting in the detection of fake shadow edges, and as a result, the real edge of the shadow remains non-exaggerated

To solve problem 1 an approach was made to restore the RGB values of the artificially-shadowed pixels with a height more than zero in the LiDAR map. Theoretically, this approach would work if the LiDAR maps correspond accurately to the RGB images. Nevertheless, besides removing the exaggerated shadows from the top of the objects, some true-shadowed pixels are removed due to false height reporting in some areas (specifically buildings edges) according to the LiDAR map as shown in Figure 15.

An approach was made to solve problem 2 by adding the option to exaggerate the shadow in more than one direction. However, this approach has solved the problem based on the results inspection. Since the same algorithm runs over the whole data set, a set of directions has to be specified beforehand to be used. Since each direction covers 45° , the images which have their shadow projected between two directions have their shadow over-exaggerated.

To conclude, the algorithm has failed to avoid the problems stated above. Implementation flaws were the reason for problem 4; it has failed to detect the entire length of the object's shadow if it contains gaps. In addition, the shadow detection method used for this algorithm detected many false-positive shadow pixels, causing problem 1. In contrast, the inconsistent shadow direction and the LiDAR height map inaccuracy in a some images in the data set were the reasons for problems 2 and 3.

Finally, the overall results of this method contained many visually unaccepted results resulting in refusing this methodology as well. Examples of some fail cases are shown in Figure 14.

5.3 Shadow map dilation

This method has been implemented with respect to the problems faced during the progressive exaggeration method. Problem 1 has been solved by filtering the pixels which have a height above the



Fig. 14. (Problem 1 corresponding to the progressive exaggeration method (Red), problem 3 corresponding to the progressive exaggeration method (Yellow), problem 4 corresponding to the progressive exaggeration method (Green))



Fig. 15. Successful shadow removal (Red), wrong shadow removal (Yellow)



Fig. 16. Top direction included in the algorithm (left), left direction included (middle), image with a correct shadow direction has over exaggerated shadow areas (right)



Fig. 17. The car is positioned in different locations in the RGB image (left) and the LiDAR map (right)

threshold from the extracted shadow map. Problem 2 was solved by detecting shadow areas regardless of their direction. Finally, problem 3 has been solved by dilating the extracted shadow map forcing the shadow gaps to disappear.

This method has shown more promising and consistent results, with few fail cases which are closely similar and have not caused a severe problem in terms of output consistency.

The inconsistencies caused by the LiDAR map are unavoidable; as shown in Figure 18, the LiDAR map does not report a height at the edges of the roof, causing this part of the roof to be dilated as it is not filtered from the shadow map before the dilation process.

The inconsistencies caused by detecting false shadows over vegetation areas are fewer throughout the whole data set due to shadow area erosion. An example of the case is shown in Figure 19.

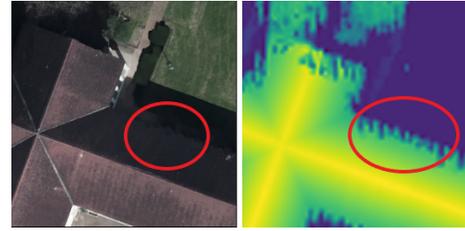


Fig. 18. Output image using shadow dilation method(left), corresponding LiDAR height map(right)



Fig. 19. Original RGB image(left), Output image without using erosion(middle), Output image with using erosion(right)

5.4 Result comparison

The results of the trigonometrical and progressive exaggeration methods were quite inconsistent and unsatisfying. However, the shadow map dilation method results have shown more consistency and realism throughout the data set.

The results examples discussed in Figures 12 to 18 have shown that the output inconsistency increases as the dependence on the LiDAR height maps in the algorithmic process increases.

The trigonometrical method was based on the height values retrieved from the LiDAR maps to calculate the objects' shadow lengths. However, the method has shown unacceptable results due to the one-to-one mapping problem between the RGB image and the LiDAR maps. The progressive exaggeration method had less dependence on the LiDAR map. It considered the height only to detect objects to be exaggerated, and it has shown fail cases when an object in the LiDAR map is in a different position in the RGB images as shown in Figure 17, but it has shown significantly better results than the trigonometrical method. The shadow map dilation method depended entirely on the shadow map extracted from the gray-scale image; it only depended on the LiDAR map when filtering the RGB pixels with a significant height. As a result, it has shown clearly more consistent and dependable results.

6 EVALUATION

The results evaluation procedure is carried out by distributing 20 surveys to 20 different students. Each survey includes five original images, their corresponding LiDAR and shadow maps, and the exaggerated output image with five questions to ask. A total of 15 surveys were collected successfully from the students and have been used for further algorithmic inspection.

The students were asked to give a rating on a scale from 1 to 5 to each of the images given a specific question. The questions aim to measure the essential metrics directly affecting the research questions. The metrics were chosen based on the shadow characteristics

such as direction, length, shape, and detected volume. The survey results can be analysed qualitatively to draw an overall evaluation of the algorithm results and a satisfaction index of each metric, in addition to knowing where to improve it. The five questions of the survey were structured as follows (on a scale from 1 to 5):

- (1) Direction: How similar is the direction of the exaggerated shadows in the output image and the real shadows in the original image?
- (2) Shape: How realistic are the exaggerated shadows with respect to the original shadow shape?
- (3) Volume: How much of the real shadow is exaggerated?
- (4) Length: Is the extra shadow length reasonable compared to the real shadow length?
- (5) Overall evaluation: How easy was it to differentiate between the original image and the exaggerated image?

The evaluation results in Table 1 below has shown a promising feedback on the shadow dilation method. A detailed analysis of the results is found in the following subsection.

Table 1. Evaluation results.

Questions	Mean score	Standard deviation (σ)
1	4.5	0.74
2	3.75	1.20
3	3.5	1.36
4	4.0	1.0
5	3.5	1.42

6.1 Feedback analysis and further improvements

After a detailed analysis of the evaluation results some minor problems were raised from the feedback:

- (1) There seemed to be a sizeable amount of non-uniformity in the generated shadows with respect to the original shadows. In some examples, the exaggerated shadows encompass a significant portion - or all - of the regions covered by the original shadows, whereas in others, only portions of the shadows are exaggerated to a noticeable degree
- (2) Some images contained exaggerated shadows on higher structures.
- (3) Some shadow areas are not detected in a few images

Problem 1 has been caused by the color distinction between the real shadow and the exaggerated shadow, as described in the shadow dilation methodology section. However, this behaviour was intended to distinguish between both types of shadows, but it has affected the realism of the exaggerated image. As a result, this has been solved by using the same shadow color/brightness of the shadow areas for exaggeration, as shown in Figure 20.

The reason for problem 2 was that even after filtering the objects' pixels from the image, some dilated shadow areas expand on higher structures. This happens when multiple objects are placed at a small distance from each other. This has been solved by swapping the object filtering and the shadow map dilation steps. This has forced the exaggerated shadow parts on the higher structures to be filtered after the shadow map dilation step, as shown in Figure 21.



Fig. 20. Darker exaggerated shadow area (left), shadow area exaggerated with the same real shadow color(right)



Fig. 21. Shadow is exaggerated on cars(left) Exaggerated shadow is removed from the top of the cars (right)



Fig. 22. Thin and scattered shadow area is not exaggerated(left), same shadow area is exaggerated(right)

Finally, problem 3 was caused because of two reasons. First, some images have their shadow areas on higher structures and are therefore filtered; this behaviour was intentional, so the RGB values of the buildings remain unchanged. Second, some shadow areas contained gaps and scatter. As a result, it has been filtered during the erosion step and therefore not exaggerated. It has been improved by smoothing the gray-scale image before extracting the shadow areas from it [1] in addition to increasing the kernel matrix convolved with the image to 9×9 and using one iteration for the dilation process. Smoothing the gray-scale image has allowed more shadow areas to be detected, and the larger matrix allowed it to be exaggerated with more intensity as shown in Figure 22.

In general, the mean scores of each question are above average, which is satisfactory. However, questions 2, 3, and 5 have significantly higher standard deviations compared to their mean scores. This means that some answers for these questions deviate more from their mean score than questions 1 and 4. This confirms that the algorithm is marginally more consistent in executing the correct shadow direction and length than the other metrics.

Problem 3 discussed earlier in this section, was one of the reasons for the lower mean and the higher standard deviation scores for questions 2 and 3, besides that some images has contained lighter shadow areas which were not detected. This issue has caused a reasonable portion of the shadow area to remain non-dilated and

therefore not exaggerated. The non-dilated lighter shadow areas in the output image have the same shadow shape, volume, and length as the original RGB image, resulting in declining scores of their related questions. To solve this problem, the threshold for detecting the shadow areas from the RGB image had to be increased without detecting more false-positive shadow pixels. However, after smoothing the gray-scale image, the threshold can slightly increase to detect brighter shadow areas and fewer false shadowed pixels. This approach was very challenging to apply as both variables directly correlate.

The last evaluation question was slightly tricky, as the answers correlate directly to the amount of shadow exaggerated in the output image. Since the data set contains diversified images from various locations and different amounts of shadow areas to be exaggerated, some images have no significant shadow areas to be exaggerated and therefore are harder to differentiate between the original image and the output image. In contrast, some images contain large shadow areas to be exaggerated and, therefore, easier to differentiate between the original image and the outcome. This explains why the standard deviation value corresponding to this question is the highest among the other questions' values.

Finally, the improvements performed in this section can help in decreasing the standard deviation values of the other questions as the corrected scores resulting from the algorithm fixing are expected to be less deviated and closer to the mean score.

7 CONCLUSIONS

Throughout this research, three methodologies were carried out to tackle the problem. However, the trigonometrical and progressive exaggeration methods have proven inconsistent and unacceptable due to implementation and data set inconsistencies as, from the results inspection, they have not satisfied the metrics used for evaluation. In contrast, the shadow map dilation method's results were more satisfactory considering all the metrics used to evaluate the problem results. Despite this adequate method, it can be improved by analysing the different outcomes using different parameter values, such as changing the number of dilation/erosion iterations and the kernel size. Besides, different techniques can further eliminate the noise caused by detecting false positive shadows. Nevertheless, the method has shown promising results in answering the research questions.

Finally, the progressive exaggeration and shadow map dilation methodologies can be combined in future work. A way to do this is by eroding the shadow map extracted from the gray-scale image and replacing the dilation process with the progressive shadow exaggeration. This will allow the shadow to be exaggerated progressively in the low noise shadow map. The process then will be carried out the same way described in the dilation methodology resulting in an exaggerated version of the image.

8 PSEUDO-CODE/S

The pseudo-code corresponding to each of the methods used is shown in this section:

Algorithm 1 Trigonometrical method pseudo-code

```

1: procedure MAIN(rgb, map, heightThreshold, coordinates, direction)
2:   map ← interpolate(map)
3:   outputImg ← copy(rgb)
4:   color ← grayColor
5:   offset ← offset(direction) (Determines the coordinates
   change given a specific direction)
6:   for pixel in rgb do
7:     x ← pixelxCoordinates
8:     y ← pixelyCoordinates
9:     objectHeight ← map(x, y)
10:    if objectHeight ≥ heightThreshold then
11:      observedShadowLength ←
   calculateLength(coordinates)
12:      sunAzimuth ←
   arctan(objectHeight/observedShadowLength)
13:      artificialShadowLength ←
   objectHeight/tan(sunAzimuth)
14:      for i = 0; i < artificialShadowLength; i ++ do
15:        if objectHeight ≥ map(x, y, offset) then
16:          outputImg(x, y, offset) ← color
17:        end if
18:      end for
19:    end if
20:  end for
21:  return outputImg
22: end procedure

```

Algorithm 2 Progressive exaggeration method pseudo-code

```

1: procedure MAIN(rgb, map, heightThreshold, direction)
2:   map ← interpolate(map)
3:   outputImg ← copy(rgb)
4:   color ← grayColor
5:   offset ← offset(direction) (Determines the coordinates
   change given a specific direction)
6:   for pixel in rgb do
7:     x ← pixelxCoordinates
8:     y ← pixelyCoordinates
9:     if map(x, y) ≥ heightThreshold then
10:      for neighbour in rgb(x, y, offset) do
11:        xs ← neighbourxCoordinates
12:        ys ← neighbouryCoordinates
13:        if not isShadow(neighbour) then
14:          break
15:        end if
16:      end for
17:      outputImg(xs, ys) ← color
18:      neighbours(outputImg(xs, ys)) ← color
19:    end if
20:  end for
21:  return outputImg
22: end procedure

```

Algorithm 3 Shadow map dilation method pseudo-code

```

1: procedure MAIN(rgb, map, heightThreshold,
   dilationKernelSize, erosionKernelSize)
2:   map ← interpolate(map)
3:   outputImg ← copy(rgb)
4:   rgbWeights ← [0.2989, 0.5870, 0.1140]
5:   grayScale ← dotproduct(rgb, rgbWeights)
6:   for pixel in grayScale do
7:     x ← pixelxCoordinates
8:     y ← pixelyCoordinates
9:     if grayScale(x, y) ≥ 60 then
10:      outputImg(x, y) ← blackcolor
11:     else
12:      outputImg(x, y) ← rgb(x, y)
13:     end if
14:   end for
15:   kernelErode ← matrix(erosionKernelSize)
16:   kernelDilate ← matrix(dilationKernelSize)
17:   outputImg ← erode(outputImg, kernelErode, anchor =
   default)
18:   outputImg ← dilate(outputImg, kernelDilate, anchor =
   [0, 0])
19:   for pixel in outputImg do
20:     x ← pixelxCoordinates
21:     y ← pixelyCoordinates
22:     if map(x, y) ≥ heightThreshold then
23:      outputImg(x, y) ← blackcolor
24:     end if
25:   end for
26:   for pixel in outputImg do
27:     x ← pixelxCoordinates
28:     y ← pixelyCoordinates
29:     if outputImg(x, y) == blackcolor then
30:      outputImg(x, y) ← rgb(x, y)
31:     end if
32:   end for
33:   return outputImg
34: end procedure

```

- [8] Doo-Ahn Kwak, Woo-Kyun Lee, Jun-Hak Lee, Greg S Biging, and Peng Gong. 2007. Detection of individual trees and estimation of tree height using LiDAR data. *Journal of Forest Research* 12, 6 (2007), 425–434.
- [9] Jianghua Liao, Jinxing Zhou, and Wentao Yang. 2021. Comparing LiDAR and SfM digital surface models for three land cover types. *Open Geosciences* 13, 1 (2021), 497–504.
- [10] Liora Sahar, Subrahmanyam Muthukumar, and Steven P French. 2010. Using aerial imagery and GIS in automated building footprint extraction and shape recognition for earthquake risk assessment of urban inventories. *IEEE Transactions on Geoscience and Remote Sensing* 48, 9 (2010), 3511–3520.
- [11] Ildiko Suveg and George Vosselman. 2004. Reconstruction of 3D building models from aerial images and maps. *ISPRS Journal of Photogrammetry and remote sensing* 58, 3–4 (2004), 202–224.
- [12] Yutaka Takase, N Sho, A Sone, and K Shimiyu. 2003. Automatic generation of 3d city models and related applications. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 34, 5 (2003).
- [13] Mustafa Turker and Emre Sumer. 2008. Building-based damage detection due to earthquake using the watershed segmentation of the post-event aerial images. *International Journal of Remote Sensing* 29, 11 (2008), 3073–3089.
- [14] Yakun Xie, Dejun Feng, Sifan Xiong, Jun Zhu, and Yangge Liu. 2021. Multi-scene building height estimation method based on shadow in high resolution imagery. *Remote Sensing* 13, 15 (2021), 2862.

REFERENCES

- [1] V Arévalo, J González, J Valdes, and G Ambrosio. 2006. Detecting shadows in QuickBird satellite images. In *ISPRS Commission VII Mid-term Symposium "Remote Sensing: From Pixels to Processes", Enschede, the Netherlands*. Citeseer, 8–11.
- [2] J.Y. Gil and R. Kimmel. 2002. Efficient dilation, erosion, opening, and closing algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24, 12 (2002), 1606–1617. <https://doi.org/10.1109/TPAMI.2002.1114852>
- [3] Megha Goyal. 2011. Morphological image processing. *IJCST* 2, 4 (2011), 59.
- [4] Stefan Hinz and Albert Baumgartner. 2003. Automatic extraction of urban road networks from multi-view aerial imagery. *ISPRS journal of photogrammetry and remote sensing* 58, 1–2 (2003), 83–98.
- [5] Mariusz Jankowski. 2006. Erosion, dilation and related operators. In *8th International Mathematica Symposium*. 1–10.
- [6] Savvas Karatsiolis and Andreas Kamilaris. 2021. Focusing on Shadows for Predicting Heightmaps from Single Remotely Sensed RGB Images with Deep Learning. *arXiv preprint arXiv:2104.10874* (2021).
- [7] Savvas Karatsiolis, Andreas Kamilaris, and Ian Cole. 2021. Img2ndsm: Height estimation from single airborne rgb images with deep learning. *Remote Sensing* 13, 12 (2021), 2417.