Journal of Theoretical and Applied Information Technology
15th August 2020. Vol.98. No 15
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3D RECONSTRUCTION USING MULTI-VIEW STEREO AND DEPTH-BASED SEGMENTATION

HUTHAIFA A. AL_ISSA[1], OMAR E. Al-SHALABI[2], MOHAMMAD QAWAQZEH[3], OLEKSANDER MIROSHNYK[4]

[1 2 3] Electrical and Electronics Engineering Department, Al Balqa Applied University, Jordan.
[4] Educational and Scientific Institute of Energy and Computer Technologies, kharkiv Petro Vasylenko National Technical University of Agriculture, Ukraine.
E-mail: [1]h.alissa@bau.edu.jo, [2]omaremadalshalabi@gmail.com, [3]qawaqzeh@bau.edu.jo, [4]omiroshnyk@khntusg.info

ABSTRACT

The current advancements in the technology of cameras especially the integration of stereo cameras into smart phones has brought ease to the process of reconstructing 3D models from different images, such broad subject could be integrated with many fields like medicine which shows the importance of studying this sub-topic of image processing. Different algorithms and techniques has been studied before for this problem where each technique has pros and cons which makes studying the previous works viable to improve or extend solutions. In this research, a general analysis of using stereo-vision will be discussed alongside with integrating the common reconstruction process with Depth-First-Search algorithm (DFS) in order to segment the object of interest from the background. The main principle of proposed method is built upon is the restriction of the angles used to photograph the object of interest, it will be limited to either 2-views denoting Front, and Back views, and 4-views consisting of Front, Back, Right, and Left views where the user has to supply these orthographic views rather than taking pictures of different angles where the program has to correlate between these images to produce a 3D model. The main goal of this paper is to show that limiting the number of input images while locking the views where shots could be taken from allows using a simple algorithm for reconstruction. To make this goal achievable, orthographic views and stereo cameras have been identified as suitable choices to simplify the reconstruction algorithm.

Keywords: *3D Reconstruction, Multi-View Stereo, Depth-Map, Elongated Image, Feature Extraction.*

1. INTRODUCTION

Image processing has been playing a vital role in solving many problems that has a visual aspect because it provides elegant mathematical solutions derived mainly from multi-variable calculus, and digital signal processing. The advancement of image processing is still growing due to its integration with deep learning algorithms [1, 2, 3]. Image processing generally treats images as multivariable functions where the parameters of this function are the spatial data, usually called pixels, and the output is the intensity of this specific pixel, for colored images instead of having one function for the whole image, a set of functions that correspond to the color system used represents the input image [3, 4, 5]. Using different image processing techniques like filtering, and feature detection makes programming reconstruction algorithms much easier but at the same time output might be prone to error because of noise that might be apparent in input images. But, generally different image processing techniques offer benefits that overcomes the limitations imposed by noise.

2. RESEARCH PRELIMINARIES

3D-Reconsturction methods are categorized into passive and active methods where the latter interferers with the object by projecting laser to it to measure the depth, then uses the extracted data to reconstruct the object, and the former method only extracts information by applying algorithms to the taken images of the object, thus reducing the number of external peripheral devices needed to do the reconstruction. Passive method will be used in this research because of the previously stated reason [6, 7, 8]. When taking photos of a specific object, we are mainly interested in some points called the features of the object. A feature is a specific compact representation of data, they usually result from visual characteristics hence the name "Visual features". A visual feature may be defined in terms of intensity, shape, size, orientation, or texture, these features will act as points of interest for the 3D-Reconsturction process because they will represent the object that we are interested in reconstructing therefore the stage where feature extraction is included is the most fundamental and crucial part of the whole process [9].

The main goal of this research is to transform a set of images into a 3D mesh that consists of points, usually known as vertices. The main idea of constructing this type of mesh is by rotating these images around a specific axis to resemble the original model. This process would make a set of applications easier such as 3D printing, where instead of manually model an object with the ordinary 3D modelling software, one could use an algorithm like the one proposed here to scan an object our of different images to produce the object as a set of vertices, later on, the produced mesh vertices could be connected using a triangulation algorithm to produce a polygonal mesh.

3. RELATED WORKS

One of the proposed methods in the field of 3D reconstruction was proposed by Ramakanth Kumar [10], his method included the use of technique by the name Speeded Up Robust Features (SURF) which is used for efficient feature detection and it's widely used in the field of 3D reconstruction. Different features in a pair of images can be detected using this algorithm, then, the undesirable points (features) are omitted with the use of epipolar geometry along with extracting a relation with respect to geometry between the two views used in the pair of images resulting in a 3D point cloud that projects the pair of images to the 3D space, these points are used to reconstruct the scene.V. Brandou [11] proposed an image acquisition technique that is adequate for small scale images using multiple stereovision camera shots of the scene with a known displacement between the shots. Stereopsis is a perception of depth from scenery, D. Scharstein and R. Szeliski. [12] proposed a method to evaluate the depth map from one stereo image (a pair of images taken from adjacent viewpoints), the basic idea relies in finding the location of a point in the pair of images. The main computational technique has four steps that can be divided into two relating to cost computation and aggregation, and the other two related to disparity computation and refinement. The cost computation determines the degree of accuracy of the match between image areas that are related to the same 3D feature. A mixture of projective-reconstruction, self-calibration, epipolar geometry, and dense depth map matching can be used to lessen the restrictions of the movement related to the camera. Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc J. Van Gool [9] proposed a general method to 3D reconstruct a scene from uncalibrated pairs of images, their method produces a scaled version of the reconstructed scene (metric reconstruction) along with extracting the color textures, a vital condition that should be met is the usage of orthographic projection. The technique starts with building correlation between pairs of images using epipolar geometry, creating an initial mesh (reconstructed 3D model) out of the two images, when adding more images for reconstruction, the algorithm refines or corrects the feature points in the previous images, next, dense depth maps are computed by treating the consecutive image pairs as stereo pairs. A contribution of their algorithm is to combine the dense depth maps that were computed earlier into a complete 3D surface, moreover, the technique proposed does not rely on initial information about the camera thus lessening the number of restrictions that is usually present in other 3D reconstruction methods, the process of calibration is done by assuring that the scene is rigid along with some constraints on camera parameters known as the intrinsic parameters, these parameters are related directly to the lens and image sensor.

4. PROBLEM STATEMENT

3D-Reconstruction as a research topic has always been a challenging goal [9], it can be explained as the process of extracting shapes, coordinates, and features such as texture from real life objects and reconstructing the extracted data as points in the 3D space. The process of 3D reconstruction can be used in multiple areas such as landmark reconstruction, underwater object reconstruction, and Magnetic Resonance Imaging (MRI) reconstruction. Throughout the research, 3D reconstruction using passive methods will be used, the choice of passive methods is picked due to the advancements in hardware as well as digital cameras.

The proposed approach relies on extracting a depth map from a stereo image, finding features from multiple images of the same object, correlating these features to the extracted depth map, and projecting these points on the 3D space. The proposed approach depends on taking pictures of the object from fixed perspectives (Front, Back, etc.) in order to simplify the process of camera calibration as it requires high quality images in order to work efficiently. The process of 3D reconstruction has challenges in all its phases starting with the procedure of taking multiple pictures of the same object, the proposed method puts a constraint on the way the images are taken to simplify the remaining phases of reconstruction, taking photos should be done either by taking pictures next to each other of the same scene following a fixed distance between each pair of taken images, or taking pictures from different fixed perspectives as accurate as possible (front, back, right, left views). The next steps involves extracting features from images and storing the coordinates of the extracted features, this phase can be done using an algorithm like SIFT or SURF, after extracting the features a correlation process between every pair of images occur where features are taken according to a given criteria to ensure that all extracted points are relevant to each other in every pair of images. After extracting the features of the given images, a depth map of every image needs to be computed, the process of estimating the depth map requires a stereo image to work efficiently. The final steps of the 3D reconstruction process involves correlating the extracted feature points with their computed depth values resulting in a 3D matrix that contains the position of each feature along the X and Y axes, with the Z value as the computed depth value. Finally, the last step involves translating the computed 3D matrices into the 3D space to reconstruct the given object/Scene.

5. METHODOLGY

The main method starts with the feature extraction phase, where the extracted points from the input image acts as a starting point as it will expand along its neighborhood to cover the object of interest, thus the choice of used method is not critical. SURF algorithm was used in this research because of its speed and simplicity. Figure 1 below shows the main process discussed above.

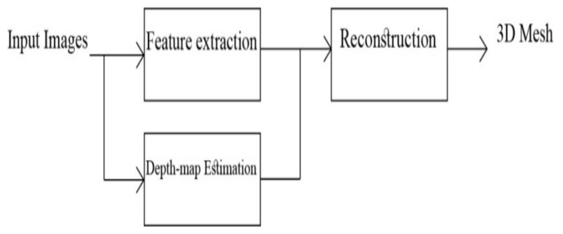


Figure 1: Main steps of 3D reconstruction

5.1 Image Photography

Since we are interested in reconstructing an object, a number of photos must be given as an input to the algorithm, the method of taking pictures of an object is the building block of the 3D reconstruction process, this research will include a method to reconstruct elongated images, and multi-view images (images taken from different perspectives), each method of taking images will affect the final reconstruction phase as it will be explained later in this article. The former method is done by capturing horizontally elongated shots of the scene, this type of photography can be used for scenes that has multiple objects to be reconstructed. Figure 2 below displays an illustration of elongated image photography.

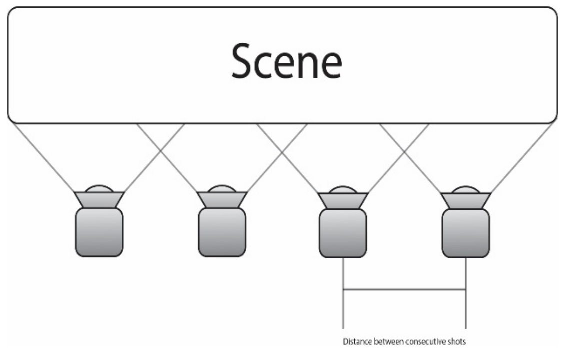


Figure 2: Elongated image method, where the distance between two consecutive shots must be near constant.

The way of taking images of the object of interest is crucial because the main method is based on restricting the angles that could be used in photography, a choice of either 2-views representing Front, and Back views, or 4-views that include Front, Back, Left, and Right views, this way, we don't have to worry about using epipolar lines to find the correlation between the input images with respect to the angles. Figure 3 below illustrates the different setups of taking pictures of an object of interest.

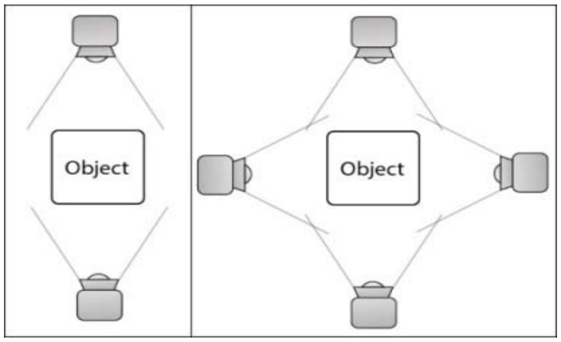


Figure 3: Different setups to take images on object of interest.

5.2 Feature Extraction And Matching

The main idea of feature extraction is to measure the variance of intensity along both axes, we refer to edges as features to be extracted, what we mean by edges is a distinctive peak in intensity compared to the neighborhood. Analogues to one dimensional signals, edges come in various shapes, it can be summarized as:

A. **Step Edge:** Where the intensity value abruptly changes from one value to another, this type of edges are easier to detect.
B. **Ramp Edge:** Where the intensity changes linearly in a certain region.
C. **Line Edge:** Where the intensity changes to a local maximum value and then returns to the original value.
D. **Roof Edge:** Where the intensity increases linearly to reach a maximum point and then decreases linearly to reach the original value.

Figure 4 below displays different types of edges discussed above. Where “Spatially” means moving in any direction in the image.

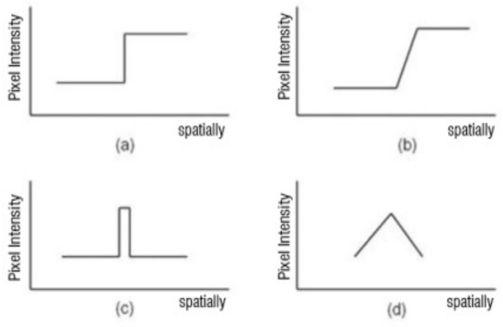


Figure 4: Four different types of edges (features), where (a) is a step edge, (b) is a ramp edge, (c) is a line edge, and (d) is a roof edge.

This following algorithm is used to identify objects in an image. Algorithm 1 below illustrates a general method to detect edges.

Algorithm 1 – Edge Detection

- Set a value for both the threshold (τ) and the standard deviation (σ) where both values should be greater than zero, (these values are used later to determine whether the pixel is a local maximum or not by comparing it to the neighborhood.)
- Compute the gradient vector $\nabla I = [I_x, I_y]$ (details in (1) and (2)).
- Compute the norm of the gradient for each pixel $||\nabla I(x,y)||$, if the norm value is greater than the threshold then this pixel is marked as an edge therefore it's considered as a feature.

The computation of the gradient vector can be done by convolving the image with a specific kernel (E.g. Sobel or pewit windows). (1) and (2) below can be used to calculate the partial derivatives of both the x axis and the y axis using the Sobel window.

$$\frac{\partial f}{\partial y} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A \quad (1)$$

$$\frac{\partial f}{\partial x} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A \quad (2)$$

Where

$\frac{\partial f}{\partial y}$: the partial derivate of an image f with respect to the y spatial axis.

$\frac{\partial f}{\partial x}$: the partial derivate of an image f with respect to the x axis.

A: input image represented as a matrix.

$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}$: Sobel window for the x spatial axis.

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \text{: Sobel window for the } y \text{ spatial axis.}$$

Where A is the matrix representation of the image. In this research, a comparison of time performance between two well-known algorithms will be done. In the next subsections a detailed description of each will be stated.

5.2.1 Scale Invariant Feature Transform (SIFT)

SIFT algorithm is known for detecting features with the ability to produce good results even if the images were scaled, hence the name "Scale invariant", other parameters that doesn't greatly affect the output are rotation, illumination, and viewport. The initial step is to create an array of scaled sub-images of the original image to ensure scale-invariance. This array of images is called a "Scale-Space". This algorithm states the steps needed to compute the scale-space of the original image. Algorithm 2 below shows how to generate a scale space.

Algorithm 2 – Scale Space Generation

- Load the image
- Generate a number of blurred out images (known as scale) by convolving the image with a Gaussian function, the formula is given by: $L(x,y,\sigma) = G(x,y,\sigma) * I(x,y)$ which is stated in (3) and (4).
- Reduce image size by a factor of two.
- Repeat the process for a number of times (this number is called Octaves).

The Gaussian function is given by (3) below.

$$G(x,y,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x+y)^2/2\sigma^2} \quad (3)$$

Where:

$G(x,y,\sigma)$: The gaussian function at x, y, and variance σ.

σ: Variance of the gaussian function, corrresponds to the amount of blurring.

Equation (4) below shows the generation of scale images.

$$L(x,y,\sigma) = G(x,y,\sigma) * I(x,y) \quad (4)$$

Where $L(x,y,\sigma)$: A blurred out image at x, y, and variance σ.

$G(x,y,\sigma)$: Gaussian function stated in (3).

$I(x,y)$: Image intensity at $x, and\ y$.

Figure 5 below illustrates the generation of scale-space.

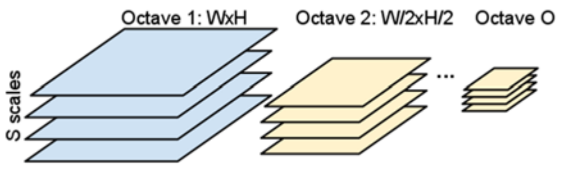


Figure 5: Generation of a scale-space, where moving vertically means blurring the image, and moving horizontally corresponds to reduction of size by a factor of two.

The next step is to use these consecutive images to extract the edges and main shapes found in the original image, we can do this by computing Laplacian of Gaussian (LOG) because it produces good results but with a major drawback of being too computationally complex, thus limiting the overall performance of the whole process. One of the solutions to this problem is by introducing a new way of finding the edges inspired by the concept of derivatives, we can replace LOG by using Difference of Gaussian (DOG) where each pair of consecutive images in each octave is subtracted, and the resulting image represents the edges of the original image. Figure 6 below shows an illustration to compute the difference of Gaussian DOG.

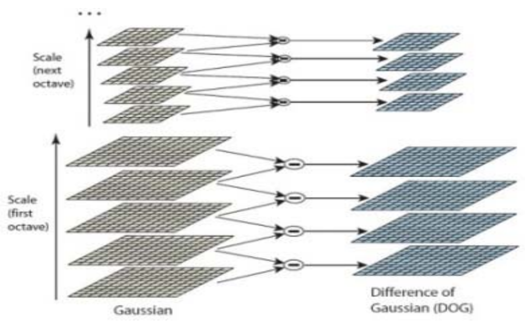


Figure 6: The method of computing DOG using subtraction.

The Generation of DOG images acts as a primer for the next steps that includes determining the local maximum and minimum points of image, these local extrema points can be treated as feature points. Algorithm 3 can be used to compute this type of points. Figure 7 shows a method to find extrema points within an image.

Algorithm 3 – Local Extrema Points Computation

- Start from the first pixel that has a neighborhood.
- Compare the intensity of the pixel to the intensities of the neighborhood pixels.
- If the value of the pixel is higher than the neighborhood pixels, mark this pixel as a feature.
- Iterate through the other pixels.

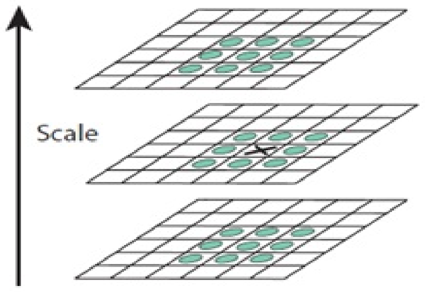


Figure 7: Method of finding extrema points by comparing the intensity value of the pixel with the neighborhood values.

Though we have extracted points that correspond to features, some of these points will serve no purpose and will be considered as noise because they could be a local extrema points but still have a low intensity value that will not correspond to an actual feature. A simple solution is to assign a threshold value, iterate through all pixels, and discard points that have a value less than the threshold. Further steps of the SIFT algorithm is to calculate the orientation of feature points, this is done by calculating the gradient of the original and blurred images and comparing it to the extracted features where the dominant orientation for each group of features is assigned as an orientation for all of them to produce a key-point, this can be done by using the gradient operator as mentioned above, the extracted orientation can be quantized into 36 bins where each bin contains 10 degrees. Figure 8 below shows the gradient magnitudes and orientations of a random input image.

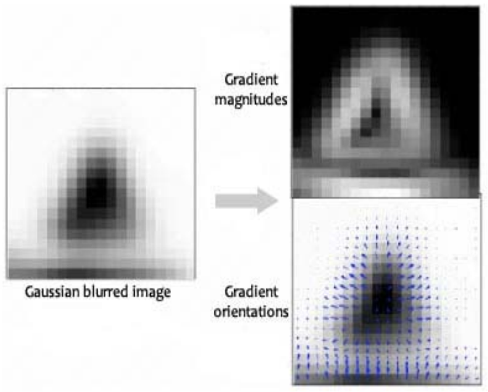


Figure 8: The computation of the gradient on the image to extract orientation of features.

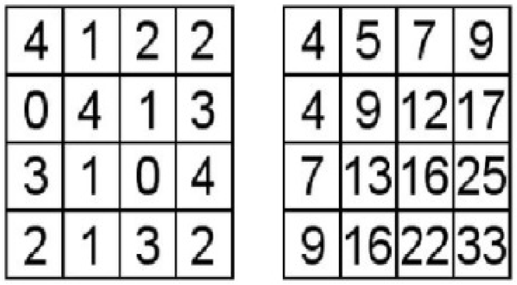
Till now, we have extracted key-points that are scale and rotation invariant, this will make any further processing done on the extracted features independent of these parameters. An additional step of the SIFT algorithm that includes matching between features in a pair of images will be explained later in this article, this step will be crucial for elongated images that contains a single object of interest.

5.2.2 Speeded Up Robust Features (SURF)

SURF is a fast feature detection algorithm that is partially inspired by the SIFT algorithm and it's claimed by the author to be more robust against image transformations. The main issue of SIFT was the performance of the algorithm is it was computationally heavy because the calculations of the difference of Gaussian can be really heavy for large inputs, the SURF algorithm uses a different approximation of the DOG of images to produce faster results. The main idea of this algorithm is to pre-compute an integral image of the original image, this will result in faster computation of box-filtering to finally compute an approximation of the determinant of Hessian. Integral images are a type of images where each pixel corresponds to the summation of all pixels above and left to it. Algorithm 4 can be used to compute an integral image. Figure 9 below shows a matrix with random values along with the integral output at the left.

Algorithm 4 – Integral Image Computation

- Load image.
- Start from the first pixel, if the pixel is on the upper left corner, produce the same value in the same location in the new integral image.
- Iterate through all pixels above and left to it.
- Sum all the pixels.
- Store the result in the same location of the pixels in the new integral image.
- Repeat for the rest of pixels



4	1	2	2
0	4	1	3
3	1	0	4
2	1	3	2

4	5	7	9
4	9	12	17
7	13	16	25
9	16	22	33

Figure 9: The original matrix is on the left, and the integral image is on the right where each pixel corresponds to the summation of the pixels above and left to it.

As a result, we can compute a box filter output with ease using integral images, the box filter is an approximation to the Gaussian function which is the average value of all the images values in a given rectangle. Next step is to compute the determinant of Hessian for the box-filtered image, this step is analogous to the subtraction of filtered images in SIFT to detect the feature points. In general, the Hessian matrix is a square matrix of second-order partial derivatives of scalar-valued functions, and it's used to describe the curvature of a function thus we can compute the determinant of this matrix to detect local extrema points. The following equation describes the mathematical model of the determinant.

$$H(p,\sigma) = \begin{vmatrix} L_{xx}(p,\sigma) & L_{xy}(p,\sigma) \\ L_{yx}(p,\sigma) & L_{yy}(p,\sigma) \end{vmatrix} \qquad (5)$$

Where:

$H(p,\sigma)$: The hessian output at point p, and variance σ.

$L(p,\sigma)$: The second order derivate of the Gaussian function at point p, and variance σ.

subscripts x, and y denotes the direction of convolution.

5.2.3 Modifications of feature extraction method

Some of the challenges that will occur in the feature extraction method include the extraction of non-related features that belong to the background thus producing a false 3D-reconsturcted model, a simple method to reduce the noise (background features) can be implemented by sliding a rectangular window along the image from the top-left corner where the number of extracted features in the sliding window will be stored, after iterating along the whole image, the coordinates of the sliding window were the number of extracted features were the maximum will be stored and used as a bounding box for the features, next is to remove every point that doesn't belong in the rectangular window, the size of the window should be large enough to be able to contain the object of interest while not including much of the background, the shape of the window will be crucial as well so different shapes will be tested to ensure the optimal shape to be used. Other modification of the feature extraction method will greatly affect the last step of reconstruction. The main issue of feature extraction is that it only extracts points that fits a certain criteria of intensity thus the number of extracted features will be very small compared to the total number of pixels in the original image, as a result, this will produce a 3D-recontructed model with lots of gaps in it that will barely visualize the original image. To solve this issue, a variation of DFS algorithm that is called Flood Fill algorithm will be used alongside with the feature extraction, the main idea is to iterate through the final extracted points and start recursively calling neighborhood pixels, if the called pixel's depth value is greater than a chosen threshold, the new pixel will be marked as a feature, then it will recursively call the other neighborhood pixels and so on, this will expand the extracted features to include the object of interest thus producing a gapless 3D reconstructed model. Algorithm 5 below shows the basic steps of Flood Fill.

Algorithm 5 – Flood Fill

- Start from the first extracted feature.
- Call the neighborhood pixels from the extracted feature pixel.
- For each called pixel, check if the intensity of the depth map for the same location is higher than a specific threshold, if it's higher, mark this new pixel as a feature, and call its neighborhood pixels, if it's lower, stop the calling procedure.

Recursion is used rather than iteration for code simplicity. Pseudo-code 1 below shows a segment of the algorithm.

Pseudo Code 1 – Flood Fill

```
• Void floodfill (int x, int y) {// x, and y are pixel indices.
• if (visited (x, y) is true or is_not_safe (x, y) is true)
• return
• if (depth (x, y) larger than threshold)
• features.add (x, y)
• floodfill (x + 1, y)
• floodfill (x, y + 1)
• floodfill (x – 1, y)
• floodfill (x, y – 1)
```

The code contains two methods called Visited(X, Y) and IS_NOT_SAFE(X, Y) to ensure that the pixel has not been visited before (so an infinite loop won't occur) and the pixel is not out of bounds respectively. The final step is to reconstruct those extracted points to resemble the object of interest in 3D space, basically, the reconstruction algorithm uses (6) and (7) to rotate the extracted points to match the perspective it was originally photographed in.

$$X' = X.Cos(\theta) + Z.Sin(\theta) \quad (6)$$
$$Z' = Z.Sin(\theta) - X.Cos(\theta) \quad (7)$$

Where:

X and Y are the original position along the x, and y axes.
X' and Y' are the rotated position along the x, and y axes.
θ is the angle of rotation measured in degrees.
Below is the general algorithm to reconstruct a 2-view stereo pair of images.

- Load features from the first image (Front side image).
- Project the features above to the 3D space.
- Load features from the second image (Back side image).
- Apply (6), and (7) to the features with a θ value of (180°).
- Project the features above to the 3D space.
- Save data as a .ply file.

For the case of 4-views, a rotation using (6) and (7) for the left and right images will cause projection distortion, a simple solution exists to treat this case. For the left view image, we swap the values between the x-axis and the z-axis resulting in a rotation of ninety degrees clockwise which translates to rotation, and for the right view image, we do the same procedure but with the addition of multiplying the z-axis values by negative one so it translates to a ninety degrees rotation anti-clockwise.

5.2.4 Feature matching

A fundamental and crucial process usually used in other 3D Reconstruction techniques like Structure From Motion (SFM) is called feature matching which takes place after the extraction of features with the aim of finding the same feature points in a pair of consecutive images, these final points are called "Correspondence points". This process will not have a major impact on the research since we're dealing with images taken from four different viewports where the angle between two consecutive shots is roughly ninety degrees. Feature matching are usually a computationally-heavy process that are used in scenarios where the camera takes different shots of the object of interest from multiple angles close to each other, in every pair of images the extracted features from one image can be projected to the other image using a mathematical matrix taken from projective geometry called Homography matrix, the basic idea of Homography is to multiply the feature point from the first image with a transformation matrix where the result will be the same feature but in the new location in the second image, the computation of Homography will not be discussed since it's used in scenarios different in the ones we're interested in. Computationally speaking, the process of feature matching is very heavy (performance wise) and it's usually done using algorithms like Random Sample Consensus (RANSAC), and Least Median Square Estimation (LMSE), both algorithms use a data structure by the name K-D tree to speed up the performance.

In this research, a simple feature matching algorithm will be used in the second configuration discussed above (elongated images) where the distance between every two consecutive shots is constant. The main idea is to do a process called image alignment which is basically stacking the pair of images on top of each other to allow further processing between the two images, the alignment process is done by drifting one image by the amount of the width of the image subtracted by the distance between the two shots, therefore, the result will be a stacked image where the intersection of the two images is the object of interest from the two images, next, we will start comparing the Euclidian distance between every feature from the first image to the features of the second image then we choose the pair of features with the minimum Euclidian distance and mark them as correspondence points. In order to reduce the number of false background features, we can use the algorithm discussed above as a prior step to the matching process. Algorithm 6 below can be used to detect the correspondence points between a pair of images after the process of image alignment. Figure 10 below shows the sliding distance between adjacent images.

Algorithm 6 – Image Alignment and Correspondence Point Detection

- Slide the second image on top of the first image by the amount of the width of the image minus the distance between the pair.
- Choose one feature.
- Calculate the Euclidian distance between the chosen feature and every other feature of the first image.
- Set the first Euclidian distance as a maximum value temporarily.
- Compare every new Euclidian distance to the previously extracted, if the new distance is smaller, overwrite the value of the distance with the new one.
- When done, delete the feature from the second image in the pair, this step is necessary to ensure there's no distortion in the reconstruction phase.
- Mark the feature, so next comparisons will ignore it, this will make the next comparisons faster.
- Iterate through the other features.

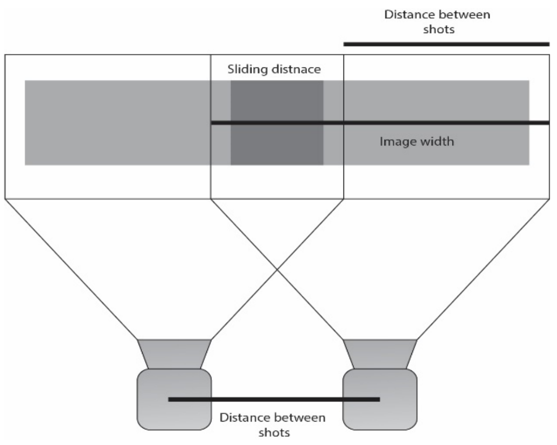


Figure 10: Sliding distance as it corresponds to the width of the image subtracted by the distance between shots.

5.3 Depth Map Estimation

The process of extracting a depth map is a quite complicated process because it involves calibrating the cameras, and some other constraints like co-planarity. Also, other techniques of depth map estimation relies on using sensors and lasers to measure the distance between the camera and the object, this is known as the active method were the former is called the passive method, which will be used in this research. This process is a field of study by its own, so a brief and simple explanation will be stated in this research about this phase. The main idea is to calculate a map named disparity from the pair of images, we will use the fact the shift of the pixel of interest and the distance between the object to the camera are inversely-proportional, so the objects near the camera will have relatively large movement compared to the objects in the background. The figure 11 below shows an illustration of the baseline and the point to be captured.

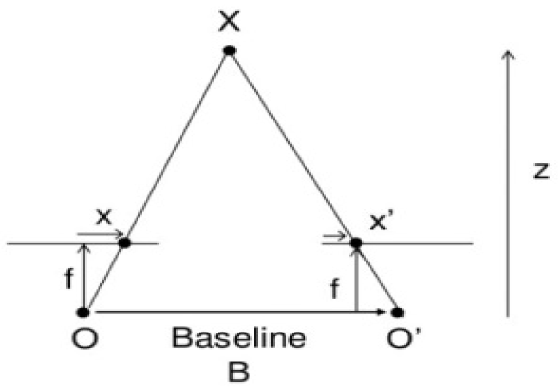


Figure 11: Horizontal distance between the point and its representation on the left and right camera x and x' respectively, f is the focal length and B is the distance between the cameras.

The implementation of this project will use Semi-Global method technique because of code complexity. The table 1 below states the main differences between the common techniques.

Table 1: Comparison of different depth map estimation techniques.

Technique	*Algorithm Complexity*	*Brief details*
Local Method	Minimum complexity between the three	Operates by matching blocks of specific sizes between the two images, assumes photo consistency assumptions (pixels of the same object in the left and right images have similar colour levels).
Global Method	Maximum complexity between the three	Relays on the smoothness of the images rather than the colours.
Semi-Global Method	Between Local and Global Methods	An optimized version of the Global method that is less computationally extensive.

5.4 Reconstruction Method

The final stage of this research is to project these extracted features into the 3D space, a simple approach is going to be used for the three proposed image acquisition methods (2-presepectives, 4-perspectives, and Elongated images). The main idea behind the three methods is manipulating the axes thus changing the direction of the extracted features. Figure 12 below shows the axes and the function of each.

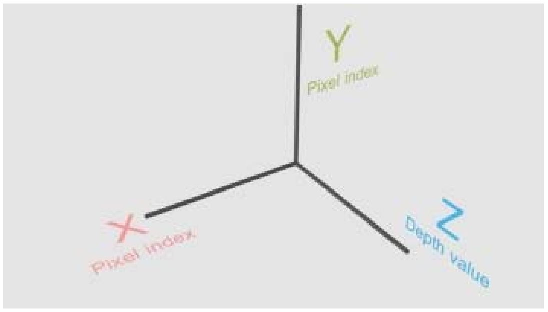


Figure 12: 3D axes, where X and Y are pixel indices, and Z is the depth value of the pixel.

To illustrate the reconstruction process, assume a picture of a cup was taken from two perspectives (front, and back), the next figures will show the output for each image. Figure 13 below shows extracted and projected features of the front image.

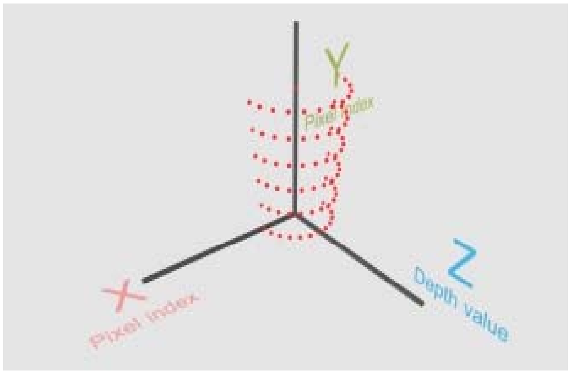


Figure 13: Extracted and projected features of the Front image.

The next step is to manipulate the axes to suit the back image, basically we flip the Z (depth) axis so the projected points will be facing the negative Z direction. Figure 14 below left side shows extracted and projected features of the back image, and right side shows the final projected features of the two images image.

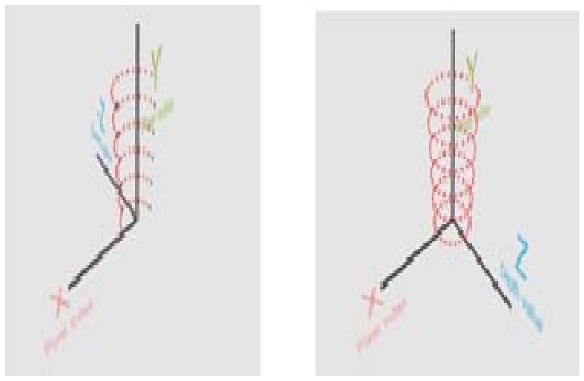


Figure 14. (left): Extracted and projected features of the Back image, (right): The final reconstructed image.

For 4-prespectives, the manipulation of axes include interchanging the values of the X and Z axes so the pixel can extend in the original X direction. The images below show the reconstruction on an object from 4 perspectives: Figure 15 below (a) shows projected features from the front image, (b) shows projected features from the left side image, with a rotation by 90 degrees clockwise, and (c) shows projected features from the back side, with a rotation by 180 degrees clockwise.

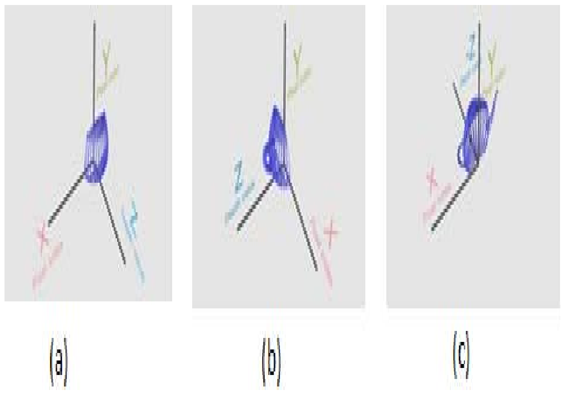


Figure 15: (a) Projected features from the front side. (b) Projected features from the left side. (c) Projected features from the back side.

Figure 16 left side below shows projected features from the right side image, with a rotation by 90 degrees anti-clockwise, and right side shows projected features from all sides which is the final mesh.

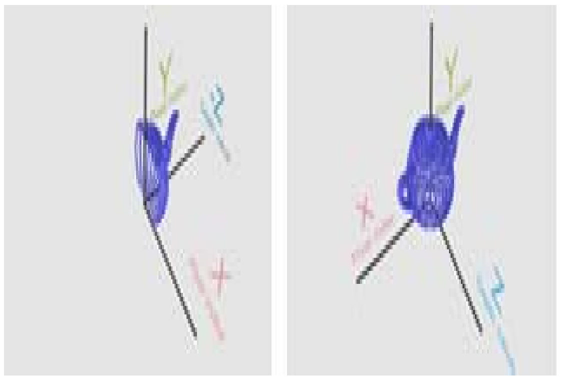


Figure 16: (left) Projected features from the right side. (right) Reconstructing a teapot from 4 images.

For elongated images, the reconstruction is basically stacking the reconstructed images next to each other. The next phase includes generating a depth-image out of the input images, no particular technique was favorited in this research, and Semi-Global methods were used because of code simplicity and performance. The generation of the depth-maps will act as a primer for the next step that involves segmentation. Figure 17 below shows an input image alongside with its depth map and depth-based-segmentation output, respectively.

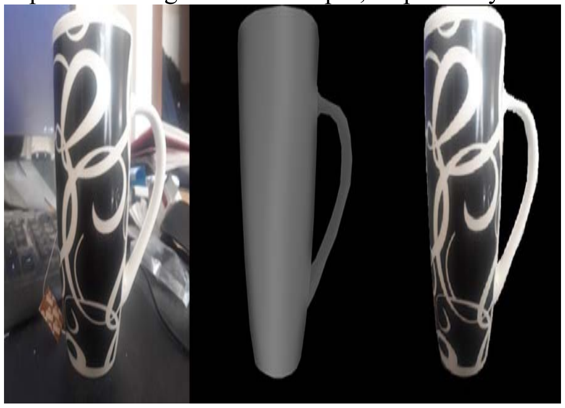


Figure 17: Object of interest on the left, manually-processed depth map on the middle, and the segmented image on the right.

6. RESULTS

The two cases that were discussed throughout the research were tested, the images used was relatively small in size where the 2-views images had the dimensions of 368 x 310 where the 4-views images had the dimensions of 413 x 310. The following results were done using the following tools: smartphone camera, Python 3.6, OpenCV 3.4.2, MeshLab 2016. Fig. 18 below shows the input and the final reconstructed 3D model.

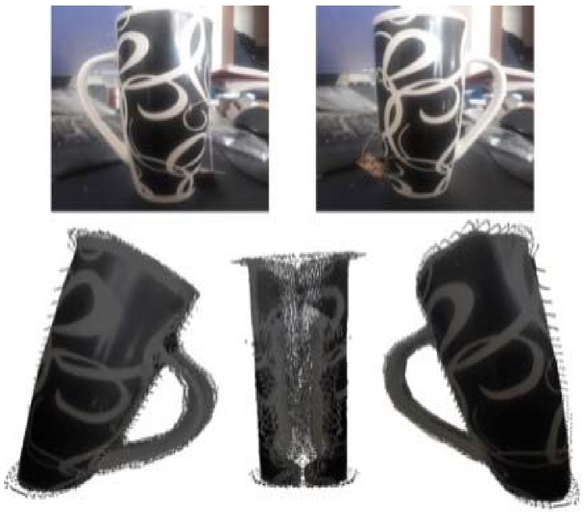


Figure 18: Input images of the 2-views case on top and the final reconstructed model on the bottom.

The four-perspective implementation included manipulation of the axes in order to rotate the extracted features in four different directions; the input images had the size of 413 X 310. Figure 19 below shows the reconstruction of the 4-views case.

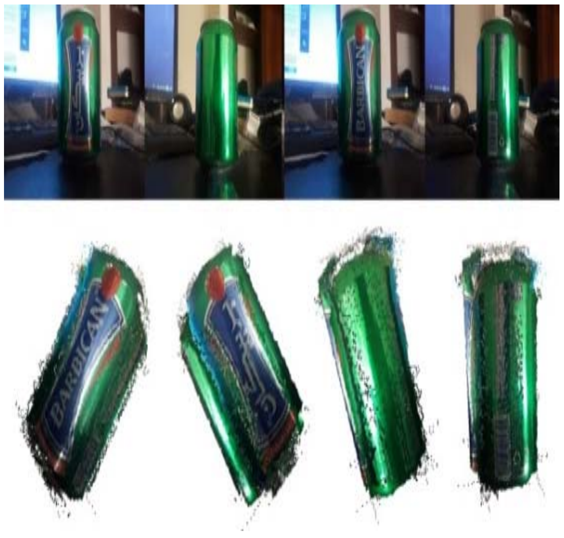


Figure 19: Input images of the 4-views case on top and the final reconstructed model on the bottom.

To compare between the results obtained in this research with similar work, the 3D reconstructed result from [9] was obtain. Figure 20 below shows the obtained result from [9].

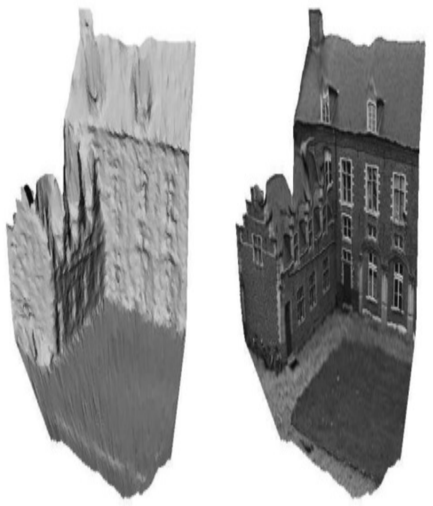


Figure 20: 3D reconstructed model from [9].

7. PREVIOUS WORK COMPARISON

The main difference between this paper and the proposed work done in [9] is that in their research a sequence of uncalibrated images is used as an input, then a step of self-calibration is initiated to construct a metric 3D model, this step requires relating the pixels of consecutive images by introducing point detectors to reduce the complexity of this step, after that the reconstruction method takes place to generate a textured metric 3D surface model. The proposed method in this paper differs by using pre-defined views to take the shots from instead of uncalibrated images which eliminates the need of computing relation between consecutive images which contributes to making the process of reconstruction easier to write.

8. CONCLUSIONS AND FUTURE WORK

As a consequence, the results obtained from the proposed method did suffer minimal noise on the edge of the 3D model because of the image size and quality, but in general it did accomplished the aim of reconstructing a real-life object using stereo pairs of images without the use of a moresophisticated algorithm like the one discussed in [9].

The scope of this subject is very broad, an integration with deep learning would be very useful, specifically integrating a neural network that can produce a depth map image out of one image rather than a stereo pair of images. Video support is another idea that could be implemented using the proposed method, instead of having input image; we could reconstruct a rigid body type of objects from a video. Ideally, the video should be rotating around the object of interest for the algorithm to accurately produce a 3D mesh out of the video. One of the challenges in this idea is to identify the object with minimal movement using feature-detection algorithms, also the depth-map estimation from a video might be complicated, but could possibly be done by measuring the difference of location between adjacent shots given that the camera is moving in a constant motion around the object. Implications of this paper include but not exclusive to using the proposed algorithm with geographical scanning, reconstruction of MRI images, and 3D printing if mesh generation was added as a subsequent step to the output of this paper. What can be understood out of the proposed algorithm is that limiting some variables like the position of the camera and the distance between the camera and the object could make the later process of reconstruction easier to write as a code without compromising the quality of the output.

REFERENCES:

[1] A. K. Jain, "A handbook of Fundamentals of Digital Image Processing", Prentice Hall of India, 1989.

[2] R. C. Gonzalez & R. E. Woods, "Digital Image Processing", 3rd Edition, Prentice Hall of India, 2009.

[3] M. Petrik and P. Stemberk, "Digital Image Processing Of Structure Response," Engineering Mechanics, 18th International Conference, Pages: 244–245, 2012.

[4] Rafael C. Gonzalez and Richard E. Woods, A text book on "Digital Image Processing," Publications of Pearson, Second Edition, 2002.

[5] Malis, E. "Visual servoing invariant to changes in camera intrinsic parameters", IEEE Transaction on Robotics and Automation, 20(1):72-81, February 2004.

[6] Allais, A-G, Brandou, V, Hoge, U, Bergmann, M, Lévêque, J-P, Léon, P, Cadiou, J-F, Sarrazin, J, and Sarradin P-M (2005). "Design of optical instrumentation for 3D and temporal deep-sea observation", Proc. of the 1st international conference of optical complex systems, OCS, Marseille, France.

[7] A. Anwer, S. S. A. Ali, and F. Meriaudeau, "Underwater online 3D mapping and scene reconstructionusing low cost kinect RGB-D sensor,"2016 6th International Conference on Intelligent and AdvancedSystems (ICIAS), pp. 1–6, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7824132/

[8] D. Tsiafaki and N. Michailidou, "Benefits and Problems Through the Application of 3D Technologiesin Archaeology: Recording, Visualisation, Representation and Reconstruction,"SCIENTIFIC CULTURETsiafaki & Michailidou SCIENTIFIC CULTURE, vol. 1, no. 3, pp. 37–45, 2015. *(2) (PDF) Computer Vision Based 3D Reconstruction: A Review*. Available from: https://www.researchgate.net/publication/336887797_Computer_Vision_Based_3D_Reconstruction_A_Review [accessed Apr 18 2020].

[9] Marc Pollefeys, Reinhard Koch, Maarten Vergauwen, and Luc J. Van Gool, "Metric 3D Surface Reconstruction from Uncalibrated Image Sequences" K.U .Leuven, ESAT-PSI, Kard. Mercierlaan 94, B-3001 Heverlee, Belgium.

[10] Ramakanth Kumar, "Analysis of Three Dimensional Object Reconstruction Method for Underwater Images" in International Journal of Scientific & Technology Research Volume 2, Issue 6, June 2013.

[11] V. Brandou. etc, "3D Reconstruction of Natural Underwater Scenes Using the Stereovision System IRIS". IEEE, June 2007

[12] D. Scharstein, and R. Szeliski, "Stereo Matching with Nonlinear Diffusion" in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96), San Francisco, June 1996, pp 343-350.