Distance and Size Measurements of Objects in the Scene from a Single 2D Image

YasirSalih[1], M.T. Simsim[1],Aamir S. Malik[2]

[1.] 1Electrical Engineering Department, Faculty of Engineering and Islamic Architecture, Umm Al-Qura University, P. O. Box 5555, 21955 Makkak, Saudi Arabia
[2.] 2Centre of Intelligent Signal and Imaging Research, UniversitiTeknologi PETRONAS, 31750 Tronoh, Perak, Malaysia
ysali@uqu.edu.sa

Abstract: Current depth estimation methods use multiple cameras, multiple images or multiple depth cues for estimating depth of field and 3D shape recovery. Therefore, these methods have large computational requirements and they generally are not suitable for real time applications which require instantaneous results such as object tracking and automated surveillance. In this paper, we employ a depth estimation algorithm from single image using trigonometry. This method uses camera's extrinsic parameters such as field of view, pitch angle and camera height. These parameters can be acquired from camera installation data and no effort is spent on computing them. Using these parameters the depth and geometry of any image point is computed using trigonometry formulas. This algorithm has very short computational time and higher accuracy compared to existing depth estimation methods which makes it ideal for real time applications. In addition, this method can compute the actual width and height of an object in the scene and as consequence the area (size) of the object is computed. Moreover, it can be used for computing distances between objects and points in the image. This can be very useful for aerial images where this method can measure the width of a river or the size of vegetation and many more.
[Salih Y., Simsim MT., Malik AS. **Distance and Size Measurements of Objects in the Scene from a Single 2D Image**. *Life Sci J* 2013;10(4):106-1119] (ISSN:1097-8135). http://www.lifesciencesite.com. 15

Keywords: Triangulation; depth estimation; 3D shape recovery; image metrology

1. Introduction

Depth or shape reconstruction is the process of retrieving 3D information of the scene given 2D images. In traditional imaging systems, 3D scene is projected on 2D imaging sensor. Thus, the depth of field is lost due to this projection. Shape recovery is a fundamental problem in computer vision and many techniques have been proposed for depth estimation and 3D shape recovery from 2D images using depth cues. Different depth cues have been used for depth estimation such as stereo cues, motion cues and monocular cues (Saxena et al. 2008).

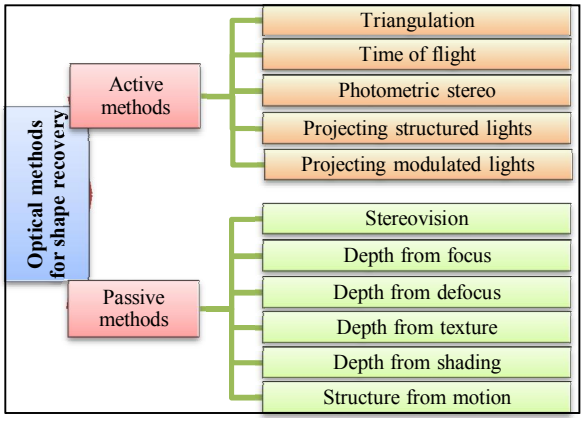

Figure1. Taxonomy of optical depth estimation/ shape recover methods

Figure 1 shows classification of optical depth estimation methods which is divided into active methods where the lighting plays an active role in depth estimation process. In the passive methods, the lighting plays a passive role in the shape recovery process. Active methods include 3D laser scanners and optical scanners where the light is projected on the scene first. Then the depth for each image point is computed using time-of-flight, phase shift of light or triangulation principles. Thus, active methods require more than just a camera for computing the depth of field.

On the other hand, passive methods use images only for computing the depth such as stereovision, structure from motion (SFM) and shape from focus (SFF). Stereoscopic methods are based on the fact that human uses two eyes in order to gauge the depth of the field. The two eyes receive slightly different view of the surrounding and stereo vision system uses these two images in order to compute the disparity map of the scene which is proportional to the depth of field (Scharstein et al. 2001). In structure from motion (SFM), the spatial disparity of consecutive frames is used to compute the depth of the scene. Objects which are closer to the camera undergo larger displacement than objects far from camera for the same amount of camera translation between consecutive frames (Lee et al. 2010). Stereovision and SFM requires calibrating the

cameras in order to match these multiple images and then computing the point to point correspondence for every new frame.

Monocular depth cues include focus, texture, motion, perspective and shading. Focus is a measure of how accurately an object is placed from the camera (Malik & Choi 2008). Objects which are out of the focused range appear defocused (blurred). Measuring the amount of focus of each point in the image, enable us to compute the depth of that point. The depth is computed using multiple images for the scene at different focus levels. Then the frame with best focus is identified for each image point and these frames indices are used to form the depth map of the scene (Lee et al. 2008). Normally large number of images is required to get better depth estimation which involve large amount of computational complexity in terms of CPU time and memory. Depth could also be computed using two defocused images. This method is known as depth from defocus which uses two images for the same scene with different levels.

Texture gradient and texture energy are also used as a depth cue; in a uniformly textured object, distance between texels varies with the depth of field which can be easily identified by computing texture gradient (Geusebroek & Smeulders 2005) (Suzuki et al. 2009). Shape from shading depends on calculating the reflectance map which is the cosine of the angle between light direction and the normal vector at each image point. A comprehensive summary of shape from shading methods was given in (Zhang et al. 1999). In addition, parallel lines that converge at infinity as well as the relative size of known objects to each others are other cues that are used in machine vision for depth perception. In recent works, multiple visual cues have been combined in order to have accurate depth reconstruction and produces realistic scenes (Hoiem & Efros 2009). However, these new methods are very complex and they require large computational complexity.

In this paper, we investigate depth estimation for real time applications such as object tracking and human robot interaction which involves retrieving multiple frames per second. Existing depth estimation techniques are not effective for real time applications because they suffer from one of the following problems:

- Active vision systems do not work with shiny or reflective surfaces.
- Stereovision requires two cameras and it has larger processing time because of the need to find correspondence points in multiple images.
- Structure from motion and shape from focus use multiple images to compute the depth of field, hence it has large computational complexity.
- In shapes from shading/texture/defocus, all methods have large computational time and poor accuracy. Moreover, these methods work only with specific type of images where the depth cue (texture, shade or blur) is clear.
- Using multiple visual cues requires higher computational complexity. Generally, supervised training is used to correlate visual cues to their depth value

As a result, existing depth estimation algorithms are not effective for real time applications because they either involve large computational complexity or they are very expensive or they use multiple images/cameras for depth computation.

In this paper, we study depth estimation algorithm that is based on the concept of triangulation using available camera parameters such as field of view, camera height and camera pitch angle. This algorithm has very short computational time and high accuracy compared to the existing depth estimation methods. Moreover, it does not require special devices and can be integrated with existing image rendering devices. The proposed algorithm can be used for depth computation using small systems such as mobile robots and smartphones.

Depth from Triangulation (DfT) method can be used for depth computation in visual surveillance which will have a great impact on the video analytics such as 3D tracking and 3D trajectory estimation. In addition, DfT method can be used for computing the ground location of any object in the scene. This is very used for generating the actual trajectory of a moving object especially for security and monitoring applications. Moreover, DfT can be used as a measurement tool to measure the actual distance between objects in the scene from singe 2D image. This feature is useful for measuring ground distance from aerial view images and navigating mobile robots.

The remaining of this paper is organized as follows. Section II provides a thorough literature survey for the existing depth estimation algorithms from single view. Section III shows the details of the depth estimation algorithm from triangulation and how to use it for distance and height measurements. Section IV, shows how object representation scheme is used in the proposed methodology. Section V demonstrates the validity of the proposed algorithm by showing some experimental results. Finally, section VI gives a brief conclusion for the work and discusses possible future extensions of this algorithm.

2. Related Works

Several authors developed algorithms for depth estimation from single image using different visual cues. Shape form shading (Wang et al. 2008; Shimodaira 2006), shape from focus (Malik & Choi 2008) and shape from texture (Kovács & Szirányi 2007) have been used for shape recovery. However, these methods have high computational complexity and they work only in images where the depth cues are uniform and prominent. Depth estimation from single image had been implemented using a combination of several monocular cues. Lila et al. (Lila et al. 2008) proposed a depth estimation method using texture and focus cues. They have employed the wavelet decomposition in order to analyze texture variations and extract focused region. White and Forsyth (White & Forsyth 2006) used texture and shading cues for computing the shape of a deformed surface. Their method computes the frontal appearance of the textured object and the irradiance map of the image in order to compute surface normal which is proportional to variations in depth.

Oliva and Torralba (Torralba & Oliva 2002) computed the mean absolute depth of the scene using Fourier spectrum of the image. This method can be used to rescale other relative depth estimation methods such as stereo and structure from motion. Chan et al. (Chan et al. 2011) used defocused cues from a single image for depth map estimation using a reverse heat equation. In this method the image is initially segmented using mean-shift segmentation then the depth is computed using a recursive reverse heat equation which deblurs the image to the optimal focus level and the relative depth is, then identified from the amount of deblurring used. Ewerth and Schwalb (Ewerth & Schwalb 2007) used motion parallax of a sequence of images for depth estimation. Futragoon and Kanongchaiyos (Futragoon 2009) used object placement information in the scene as a constraint for computing its depth of field. Prior knowledge about the object location and size in the scene are used to infer the depth of the object.

Lin and Chin (Lin & Chin 2005) developed a system that converts a 2D image into stereoscopic 3D effects. The system consists of image segmentation using online ICA mixture model, depth estimation and shift algorithm to generate the stereoscopic effects. The depth is computed by detecting the focused planes in the image and then objects at the bottom of the plane will be assigned a smaller depth value whereas objects at the top of the plane will be assigned higher depth value than the focused plane. Nagahara et al. (Nagahara et al. 2008) proposed a depth estimation algorithm from line scan panoramic images. This method allocated depth values based on color drift of image points. Color drift is the change between the RGB color components of each image point due to the camera motion (horizontal scan). Park et al. (Park et al. 2008) constructed 3D face from a single 2D image by estimating the pose of the face from the locations of the facial landmarks in the image.

Some works focused on computing the depth of the scene by classifying the image into geometrical classes using supervised learning methods. Jung and Ho (Jung & Ho 2010) estimated the depth by classifying image components into four categories (plane, cubic, sky and ground) using Bayesian learning then assigning a suitable depth value to each segment. Hoiem et al. (Hoiem et al. 2007) developed an automatic image pop-up system by classifying the image into geometric classes. Image pixels are labeled into sky, vertical and ground using supervised training. The 3D model is later created by popping up regions with vertical labels on the ground segment. Cornelis et al. (Cornelis et al. 2006) developed a 3D city model using both fast dense stereo and real time SfM algorithms. The 3D reconstruction is combined with a detection algorithm in a cognitive loop so that the object detection guides the 3D reconstruction work while the 3D reconstruction provides the object detection with scene geometry.

Gould et al. (Gould et al. 2009) decomposed the scene into semantic regions by employing a unified energy function. These regions are later used for 3D reconstruction of the scene using prior knowledge about the object classes. Hedau et al. (Hedau et al. 2009) worked on recovering the spatial layout of a room from single view by means of modeling the room components as 3D box which is computed using vanishing lines. The orientation and location of the room components are computed with respect to the room geometry. Liu et al. (Liu et al. 2010) estimated a rough depth map of the scene using a multiple stage classification for 15 types of scenes using support vector machine learning. The depth cues are extracted in form of texture information using Gaussian derivatives. Kuo and Lo (Kuo et al. 2011) developed a monocular cues approach for depth estimation of outdoor images based on multi-resolution processing. The image is segmented into coherent regions and initial depth is assigned to each of these regions in three different resolutions.

Saxena et al. (Saxena et al. 2009) worked on computing the depth of field directly from image cues using Markov random field (MRF) models. They analyzed the relationship between image features at different scales and the depth of field using a multiple scale Markov random field. In (Saxena et al. 2008) a fixed size segment is

considered and its depth is inferred using Gaussian as well as Laplacian Markov random fields. In (Saxena et al. 2009) the depth of field of a non-regular size segments (super-pixels) is estimated using Markov field learning which analyze image features as well as connectivity and occlusion between the image segments. Das et al. (Das et al. 2009) worked on improving the algorithm in (Saxena et al. 2009) by designing a new omnidirectional high pass filter that can capture more depth features than the original filters used in (Saxena et al. 2009). Although these works produced promising results for absolute depth estimation, they are very complicated and do not consider the context of the image but rather they learn the visual cues presented in the scene without semantic knowledge. Liu et al. (Liu et al. 2010) proposed a semantic labeling approach for computing the 3D structure of the scene. Semantic labeling guides the 3D reconstruction process by enforcing depth geometry constrains for some part of the image.

Parallel lines in the scene vanish at one point in the image; this method has been employed for estimating the depth from single image. Vanishing lines can be computed using Hough transform (Criminisi et al. 2000). Criminisi et al. (Criminisi et al. 2000) computed distance between objects in the scene using vanishing lines from single view. This method can be used to compute the distance between two points in the image by employing the geometry of parallel lines in the scene. Rother et al. (Rother et al. 2007) used casual people motion to compute the horizon of the scene using three different observations for the same object. These different observations for the same object were fused in one image to produce three horizon points. Barinova et al. (Barinova et al. 2008) used vanishing lines for reconstructing the surface of urban scenes. The method assumes the scene is composed of ground and vertical walls and they try to locate the ground vertical boundaries in the image which can completely define the scene structure. Reibeiro and Hancock (Ribeiro & Hancock 1999) presented a method for pose estimation using two vanishing points computed from textured planes. Vanishing points are computed from the spectral angle of textured plane. Horry et al. (Horry n.d.) used five rectangles which are centered on the vanishing point to model the background of outdoor images. Then background subtraction process was used to locate foreground object in the image and hence compute their location in the constructed 3D model. Mendonca and Kaucic (Mendonça & Kaucic 2008) used vanishing points for computing compressor angle of a jet engine. Peng et al. (Peng et al. 2010) presented 3D metric from single uncelebrated image using orthogonal vanishing points where the ratio of lines orthogonal to the vanishing lines in the image are used to infer the depth. Pribyl and Zemcik (Pribyl et al. 2011) used the size of known objects in the image (e.g. traffic signs) to calibrate the scene and measure distances and areas using the geometry of these known objects. Wang et al. (Wang et al. 2002) developed a measurement model from a single image using two orthogonal vanishing lines. Lee (Lee 2012) developed a method for height estimation from vanishing points using genetic algorithm optimization. The method requires one time calibration using a checkerboard box and it does not requires intrinsic or extrinsic camera parameters. Lalonde et al. (Lalonde et al. 2012) used triangulation algorithm for 3D reconstruction of objects in the rearview of vehicle camera. The method is based on detection of interest points and multiview triangulation. This method was fully implemented on a parallel SMID array processor.

3. Depth from Triangulation

In this paper, we employ a depth computation method that is based on triangulation in a passive manner. The proposed method works for cameras that are looking downward with a known pitch angle and height. This case is typical for surveillance cameras in order to cover a wider area and avoid occlusion from background objects. Figure 2 shows a camera setup where the area viewed by the camera is highlighted in green trapezium area. The camera is installed at a height (h) from the ground with a pitch angle (θ) w.r.t the vertical axis. The field of view of the camera is (FOV_H) in the horizontal direction and (FOV_V) in the vertical direction. Note, some cameras have similar vertical and horizontal field of view. For zoom cameras, the field of view is given as a range of maximum field of view angle, when the image is zoomed out and minimum field of view angle, when the image is zoomed in.

The size of the area covered by the camera depends on three parameters; the camera height, pitch angle and camera field of view. For example, the larger the field of view angle means it covers larger area. For zoom cameras, the field of view changes only when changing the zoom, which allows it to cover small or large scene. Increasing the camera height covers larger viewing area and decreases the per pixel resolution. Increasing the pitch angle increases the viewing area and increases the depth of field as well. The pitch angle (θ) is constrained by the trigonometry relationship of Equation (1). If the pitch angle exceeds this constraint, the trigonometry relationship cannot be maintained, and thus we might obtain erroneous results for some image points. The image has width (W) and height (H). The resolution

of the scene depends on the image resolution as well as the size of the area covered by the camera. The higher the image resolution, the finer the image element and thus it can have more accurate localization of the object location. However, if the size of the covered area is large (the camera height is large or the field of view is large), the resolution is reduced and the pixel element will be larger in size. Thus, there is a larger quantization error. The parameters mentioned earlier are generally known for all cameras and they are tuned during the camera installation process.

$$\theta < 90 - \frac{FOV_V}{2} \quad (1)$$

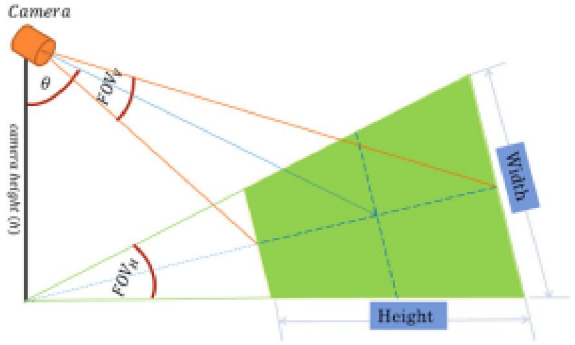

Figure 2.Typical camera installation setup.

3.1 Depth from Triangulation (DfT)

The geometrical structure of Figure 3 can serve as a base for computing the depth using trigonometry. In Figure 3, there is an object located at $p(i,j)$. In the image, the object is identified by its bottom point (feet location). Section IV gives more details about the bottom point representation and how to detect it in the image. The object in Figure 3 is located at pixel $p(i,j)$ in image coordinate. i represents the X-axis component for the object and j represents the Y-axis (height) component for the object. Since i and j are pixel coordinates, they can easily be extracted from the image. Now the object at point $p(i,j)$ can be described by the camera height (h), a rotation angle (ϕ) and a vertical angle (ψ), similar to spherical coordinates representation. The rotation angle (ϕ) is computed using the X-axis element (i vector). The angular step in the horizontal direction is defined as the change in the rotation angle due to one pixel change in the horizontal direction $(\Delta i = 1)$. From Figure 3, the maximum change in (ϕ) is (FOV_H). Since the image width is (W), the angular step in the horizontal direction is given as $(\frac{FOV_W}{W})$. Now the rotation angle is calculated using Equation (2). This angle is negative if the object is at the left side of the image and it is positive if the object is at the right side of the image.

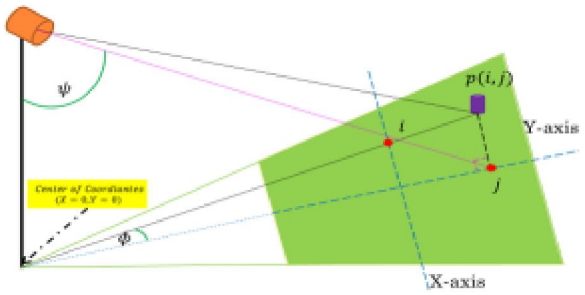

Figure 3. Trigonometry model of an object in the scene at location $p(i,j)$.

$$\phi = \left(i - \frac{W}{2}\right) \times \left(\frac{FOV_H}{W}\right) \quad (2)$$

Similarly, the angular step in the vertical direction is defined as the change in the vertical angle due to one pixel change in the vertical direction $(\Delta j = 1)$. From Figure 3, the maximum change in (ψ) is (FOV_V). Since the image height is (H), the angular step in the vertical direction is given as the vertical field of view divided by image height $(\frac{FOV_V}{H})$. Now the vertical angle (ψ) of the object at location $p(i,j)$ is computed using Equation (3). In this equation, the angle is shifted by (θ) because the central element of the image $p(\frac{W}{2},\frac{H}{2})$ has a vertical angle $\psi = \theta$ which is the installation angle for the camera.

$$\psi = \theta + \left(\frac{H}{2} - j\right) \times \left(\frac{FOV_V}{H}\right) \quad (3)$$

Given these two angles the vertical ground distance between an object located at point $(p(i,j))$ and the camera pole is computed using Equation (4).

$$Y = h \times \tan(\psi) \quad (4)$$

Now the horizontal distance X (Figure 4) is computed using Equation (5). The distances$Y$ and X represent the ground location of the object $P(X,Y)$ assuming that the center of coordinates is beneath the camera directly. The ground location tells the actual location of the object in the scene in real world coordinates.

$$X = Y \times \tan(\phi) \quad (5)$$

The depth of field (Z) represents the diagonal distance between the camera and the object of interest $p(i,j)$ and it is computed using Equation (6). The ground location tells where the object is located in the ground and the depth of field shows how far is the object from the camera location. Although these two descriptions are not independent as the depth of field is proportional to the ground location, but it gives more insight for localizing the object which could be used for further analysis.

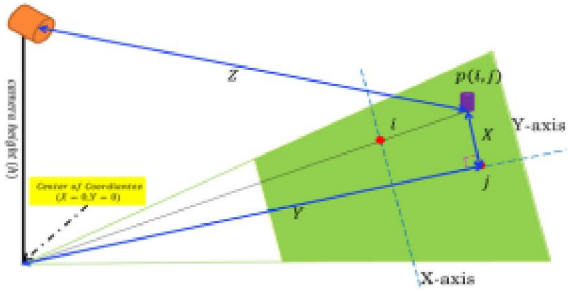

Figure 4, Computing the ground location and depth of field for an object at location $p(i,j)$.

$$Z = \sqrt{h^2 + Y^2 + X^2} \qquad (6)$$

This algorithm is known as passive triangulation because triangulation is also used for depth computation with active depth sensors such as laser scanners whereas in this implementation only the image is utilized without any additional hardware. Figure 5 shows a complete flow diagram for depth from triangulation method. This method takes seven inputs which are namely the camera height (h), the camera vertical angle (θ), the fields of view (FOV_H *and* FOV_V) and the object location in image coordinates $\{p(i,j)\}$. Firstly, the algorithm computes the horizontal rotation angle (ϕ) and the vertical pitch angle (ψ) for the said object. Then, Equation (4) and Equation (5) are used for computing the ground location $P(X,Y)$ of the object and Equation (6) is used for computing the depth of field (Z) for the object. Thus, given an object image coordinates this method return its 3D coordinates with high accuracy in a very short computational time. Compared with the methods discussed in Section II, this method has the minimum computational requirement. In addition, this technique returns the absolute coordinates of the object while other methods return relative depth of field.

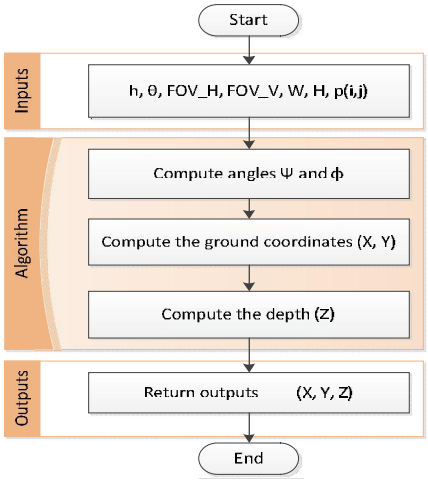

Figure 5. Flow diagram for geometry from triangulation methods.

3.2Distance and Height Measurement

Triangulation algorithm can be used to compute the actual distance between two points in the image. This can be very useful in aerial images where the pilot can choose two points in the image and the DfT algorithm measures the exact distance between them. In addition, DfT method can also be used to compute the distance between two detected objects in the scene. This is used in automated surveillance for detecting unattended object in the scene by measuring the distance between the object and the person who left it. In Figure 6, there are two objects in the scene; one at pixel $p_1(i,j)$ and another object at pixel $p_2(i,j)$ in the image. The ground location and depth of field for the first object is computed by the DfT as$p_1(i,j) \leftrightarrow P(X_1, Y_1, Z_1)$ and for the second object is computed as $p_2(i,j) \leftrightarrow P(X_2, Y_2, Z_2)$. Then the distance between these two points is computed using Equation (7).

$$D_{p_1-p_2} = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2} \qquad (7)$$

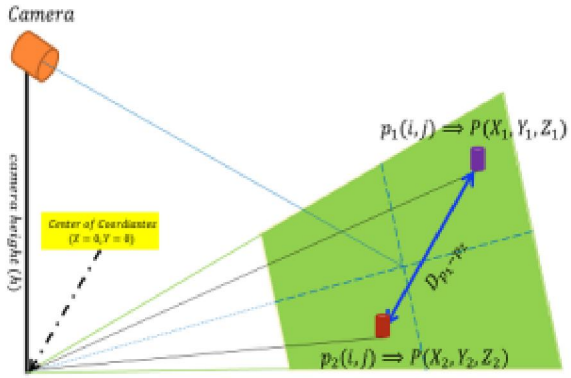

Figure 6. Measuring the ground distance between two objects in the scene using DfT algorithm.

$$X = h \times \frac{Y2 - Y1}{Y2} \qquad (8)$$

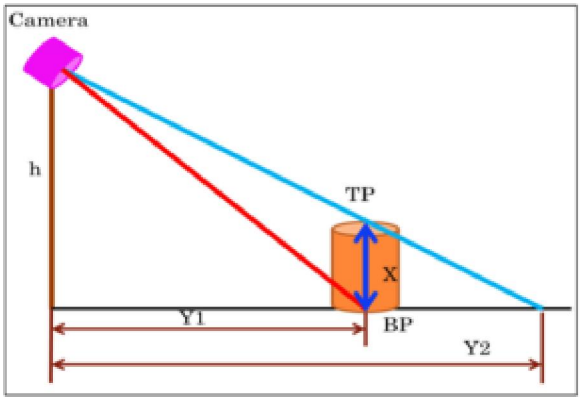

Figure 7. Computing the height of an object from a single image using depth from triangulation.

4. Object Detection and Representation

Triangulation methods compute the depth of field as well as the 3D ground coordinates for the objects given a point in the image and a known camera setup. However, objects in the scene occupy a patch of pixels and not one point. Normally, centroid point is used to represent an object with a single point because it is much easier to detect the centroid of an object than detecting its outer points. Moreover, centroid is robust to mathematical morphology which is normally used to eliminate noisy pixels.

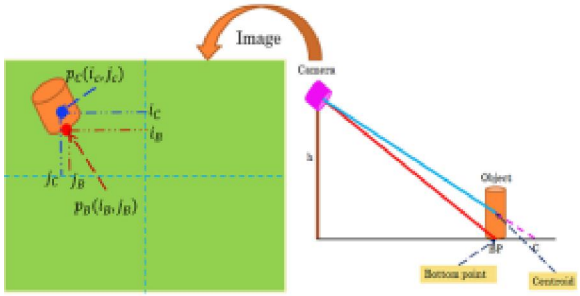

Figure 8. An arbitrary object in the scene represented by its centroid as well as bottom-most location.

In this implementation, the object of interest is represented by its bottom most location as it was stated in the Section III. Let's consider the example in Figure 8 to explain the reasons of this selection. This figure presents a sectional view of an object in the scene where the camera is installed at a known height and with a known vertical angle. The rightside of the figure also shows the image captured by the camera for that object. Let's assume that the object is represented by its centroid location$p_C(i_C, j_C)$. From the sectional view (right figure), $p_C(i_C, j_C)$ represents the location of an object standing at point (C) in the scene which is not the true location of the described object. Now let's consider the other scenario where the object is represented by its bottom-most (feet) location in the image which is point $p_B(i_B, j_B)$. In the scene, the bottom point is the true point at which the object stands (point (BP) in the right figure). As a result, in order to compute the true location and depth of field of any object in the scene, its bottom-most location in the image must be considered rather than other forms of point representation such as centroid point.

4.1 Non-flat Surface

The proposed method is developed for depth computation on flat surfaces where the object of interest and the camera pole are assumed to lie on a flat ground. Let's consider the example in Figure 9. The true location of the moving object is at point (A) while it is seen in the image at point (B) of the ground which is not the true location of the object with respect to ground. Therefore, the estimated depth will be larger than the actual depth of the object. If the height of the uneven surface is known, the true depth could be computed by taking the camera height from the tip of the uneven surface. Thus, this method can be implemented on flat surfaces as well as non-flat surfaces with a known height.

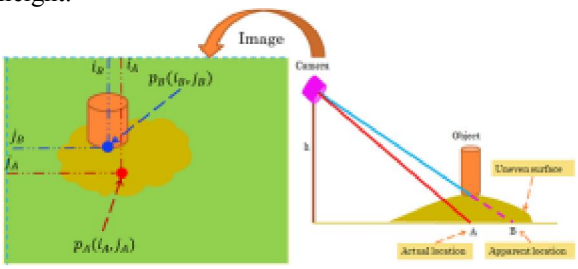

Figure 9. Detecting an object location on a non-flat surface.

4.2 Shadow Effect on DfT

Cast shadow gives false location for the moving object which affects the depth computation algorithm. Therefore, shadow/highlights must be eliminated before computing the depth. In Figure 10, the moving object is at location (B). If the object detection algorithm falsely detects the shadow as part of the object; this object will be detected at (A) which is not the true location. This indicates that it is necessary to eliminate shadow effects before identifying the object location in the image. Many techniques have been proposed for shadow elimination in the published literature. Xu et al. (Xu et al. 2005) assumed that shadowed regions maintain the same colors and texture properties as non-shadowed one. Thus shadow regions can be detected by comparing its color and texture with the non-shadowed background if the background is known or can be estimated. Branca et al. (Branca et al. 2002) used the photometric gain which is the ratio between the luminance of current frame and the background luminance for detecting shadow regions in the foreground image. Shadow regions tend to have photometric gain less than 0.90.

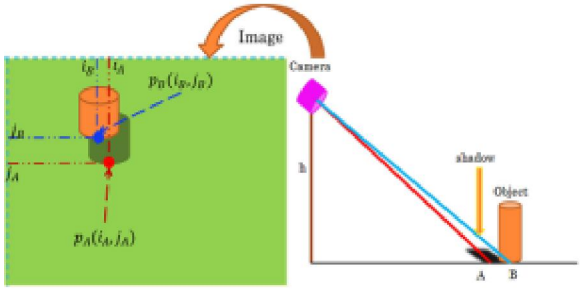

Figure 10. Shadow effects on object detection.

4.3 Occlusion Effects on DfT

If an object is occluded by the background or another object partially or fully, it cannot be seen by the camera at its correct place. Therefore, the true object location cannot be detected in the image. Usually, the occluded object location is estimated based on its motion history. For example, if the velocity and direction of the moving object is known, the new location of the moving object can be interpolated from the previously known location using the motion history of the object. In Figure 11, Object 1 partially occludes Object 2. Therefore, the location of object 2 in the image cannot be directly obtained but rather it can be estimated using knowledge about its previous location and motion vector and velocity.

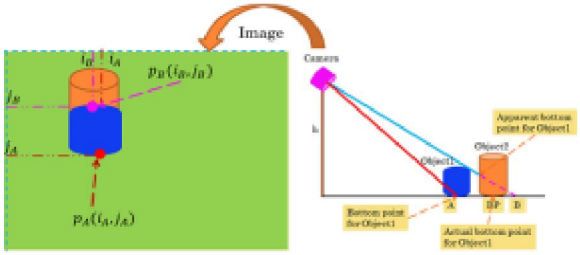

Figure 11. Occlusion effects on object detection.

5. Experimental Results and Analysis

5.1 Data Collection

There is no special procedure for data collection with the depth from triangulation (DfT) method. However, the basic camera information has to be known prior to data collection. This includes the camera height, camera pitch angle and camera fields of view. The pitch angle is setup so that a valid triangulation is maintained which means that the vertical angle for any point in the image must be less than 90 degrees according to Equation (1). This angle is set near the maximum value in order to have a wider view of the scene.

Firstly, we conducted experiment to measure the quantization error for different cameras with different specifications. Quantization error is the maximum error incurred due to one pixel miss-detection in the image. This error is computed by computing the difference between the depth of field of the selected point and the furthest point among its 8-neighboring pixels. The quantization error is helpful in analyzing the performance of the depth estimation method since no system can exceed the accuracy of its quantization level. Table 1 shows a list of image capturing devices with their basic information. The camera height is set at 10.0m for all cameras and the pitch angle is adjusted so that the maximum depth of field is 150.0m for all cameras. Image size and field of view information are acquired from cameras specification manuals provided by manufacturers.

Fig.ure 12 shows the quantization error for cameras listed in Table 1. The figure shows depth of field distances from 10.0m to 150.0m. In these graphs, the error is highly influenced by the image resolution and the field of view. High resolution imaging devices have smaller error; for example Canon 1000D camera and N900 smart phone have quantization error of less than 0.5m for a distance of 150.0m from the camera. While surveillance cameras have lower resolution (320x240 is a typical one); hence the quantization error is higher. For example Samsung cameras have error of 2.0-2.5m for SDZ-375 and SNB-300 respectively. The Dlink camera has the largest error among all the selected cameras (5.0m at 150.0m distance) because it has the lowest resolution.

Table1. Data collection using different types of cameras.

Camera	Camera model	Image size	Field of view	Camera height	Pitch angle
Cam1	Canon 1000D	3888x2592	64.30^0x 45.30^0	10.00m	63.0^0
Cam2	Samsung SDZ-3750	704x576	55.5^0x4 2.5^0	10.00m	64.9^0
Cam3	Samsung SNB-3000	640x480	35.67^0x 26.87^0	10.00m	72.8^0
Cam4	Dlink DCS-2120	320x240	49.60^0x 37.20^0	10.00m	67.6^0
Cam5	N900 camera	2584x1938	44.90^0x 33.67^0	10.00m	69.4^0
Cam6	Logitech Webcam	1600x1200	60.00^0x 45.00^0	10.00m	63.7^0
Cam7	Nao robot camera	640x480	27.84^0x 20.88^0	10.00m	75.8^0

On the other hand, the larger the field of view the larger is the quantization error because the camera covers large area and thus the area covered by one pixel is larger. In Table 1, Cam#2 and Cam#7 have the same resolution but Cam#7 has smaller error because it has smaller field of view compared to Cam#2. In addition, the quantization error is also influenced by camera height and the pitch angle. Since this study focuses on depth estimation for visual surveillance where the camera height and pitch angle are fixed, the camera height and pitch angle have been adjusted so that image cover a depth of field up to 150m for various cameras with different resolutions and fields of view

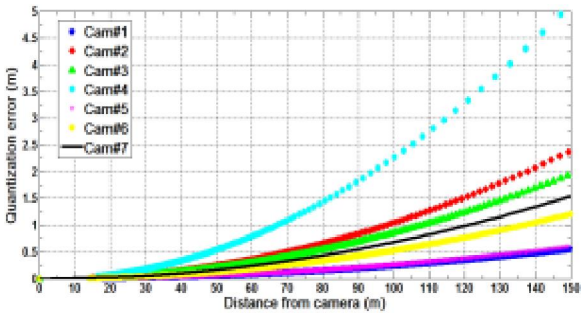

Figure 12. Quantization error for different types of image capturing devices.

5.2 Results and Discussions

Depth from triangulation method has been used for measuring depth of field and ground location of points of interest in the image. In addition, DfT is used for measuring the size of an object in the scene (width and height) as well as measuring the distance between different points in the image. Firstly, let's examine the method's ability to measure depth of field for selected points in the image. Table 2 shows 10 depth of field measurements computed using DfT and its associated actual measurements. The images in this experiment were captured using the first camera in Table 1 (Canon 1000D) where the camera was placed at a height of 10.48m and with viewing angle of 67.0^0. Figure 13 shows 10 points selected at the distancesfrom the camera. The actual measurements have been acquired using laser rangefinder which measures the distance from the point of interest to the camera center. The fourth column in Table 2 shows the absolute error while the last column shows the error percentage with respect to ground truth measurements. In Table 2, the maximum error is 3.6m at the distance of 150m. This error is only 2.5% of the actual measurement and moreover, at this distance the quantization error is large while the point of interest in the image is not very clear at this far distance.

Figure 13. Points at which the depth of field data are computed

Table 2. List of depth of field measurements using DfT and corresponding ground truth measurements

Case	Ground truth (m)	Estimated depth (m)	Error (m)	Error (%)
1	17.37	17.31	0.06	0.35
2	24.94	24.77	0.17	0.69
3	37.54	37.24	0.30	0.80
4	49.61	48.61	1.01	2.03
5	68.84	67.24	1.60	2.32
6	80.81	79.09	1.71	2.12
7	105.67	102.91	2.76	2.61
8	129.92	126.92	3.00	2.31
9	133.99	130.79	3.20	2.39
10	144.54	140.94	3.60	2.49

In Figure 14, the error in Table 2 is plotted and compared with quantization error. This error represented the smallest mesureable error at the point of interest in the image. Therefore, it is important to comapre the measurment error with the quantzation error in order to highlight the accuracy of the measurement. Generally the measurment error obtained increases with the distance from the camera. But this increase in error is not necessarily monotonic because the error in pixel selection is random. In Figure. 14, the maximum error obtained is 3.6m at a distance of 150m while the quantization error at this distance is 0.5m which means the error is due to multiple pixels dispalcement. This is reasonable because the object at that distance (as shown in Figure. 13) is not very clear and it cannot be precisely located in the image.

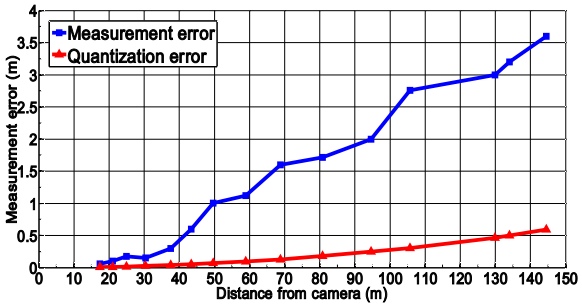

Figure 14. Graph of error in depth estimation with varying distance from the camera

The same experiment was repeated for the image shown in Figure 15, where 10 points were identified in the scene and their depth of field were computed using DfT method. Unlike the previous image (in Figure 13), the ground is not flat. Therefore this image is expected to have lower accuracy compared to the previous one.

Figure 15. Image of an uneven surface with 10 points selected for depth computation.

Figure 16 shows the error in computing the depth of field for various distances and compares it with the quantization error at these points. In Figure 16 the maximum error is 5.5m at a distance of 60m from the camera, which is significantly large. This is mainly because the surface is not flat. In the previous experiment the error at 60m was only 1m while in this experiment it is more than 5m for the same distance because of differences in camera height and viewing angle.

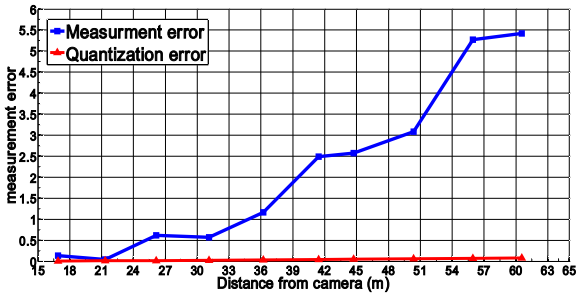

Figure 16. Graph error in depth estimation with varying distance from the camera for non-flat surface.

5.3 Height Measurement

In this subsection, we show the implementation of DfT for measuring the height of an object in the scene. To measure the height of the desired object, its bottom-most and top most points in the image are selected. Figure 17 shows height measurement for each of the two persons in three different images. The images in the first row were captured using Dlink DCS 2120 camera. The height of camera is 2.827m, the vertical angle is 60.0 degrees and with resolution of (320x240). The images in the second row were captured using Canon 1000D camera installed at a height of 1.913m and with pitch angle of 69.0 degrees.

Figure 17. Height measurement of a two person in different images.

In the first row of Figure 17, the true height of the person is 1.70m while it is measured as 1.717m, 1.697m and 1.590m respectively at four different distances from the camera. In general the measurement is very accurate except for the last image where the error is almost 10cm. This is because at this distance the person is far from the camera and in addition to that, part of the person head is not visible in the image. In the second row, of Figure 17 the true height of the person is 1.76m. In the subsequent images the measured height of this person was 1.760m, 1.7594m and 1.7694m. In the four images, the maximum error was less than 1cm which is considerably high.

Similarly, DfT was used for measuring the height of two boxes in several images. In Figure 18, the first row shows images taken for the first box using DCS 2120 camera installed at 1.292m and the pitch angle is 55.0 degrees. The actual height of the box in the first row is 0.291m while the measurements indicated similar height in multiple images and from different points. The maximum error obtained was in the rightmost image of the first row where the measured height is 0.2779m which is only 13mm less than the actual height. The second row shows images for a second box taken using Logitech webcam installed at height 1.879m and with pitch angle 60.0 degrees. The actual height of the box is 0.516m. In all the images of Figure 18, the measured heights of the box are close to the actual height. The largest deviation was recorded in the third image where the measured height was 0.528m which is only 12mm different from the actual height of the box. In general, the measurements recorded for the second row of Figure 18 have higher accuracy than the one obtained for the first one because the image in the second row has higher resolution than the one in the first row of Figure 18.

Figure 18. Height of a box measured from different views and at different sides.

5.4 Width Measurements

This algorithm can also be implemented for measuring the width of objects of interest in the scene or even measuring the actual distance between two points in the image. The width is computed by selecting two points at the edges of the object of interest. Then the 3D location is generated for these two points. After that, the width or the distance between these two points is computed using Euclidean distance measure. Figure 19 shows width measurement for two boxes. The images in the first row were captured by Logitech webcam installed at a height of 1.879m and 60 degrees vertical angle while images in the second row were captured using Dlink DCS2120 camera installed at a height of 2.827m with 60 degrees vertical angle. In the first row, the actual width of the box is 0.571m. The width of the box is measured at the bottom side by selecting two points then using Equation (9) to measure the width. At the top side of the box, the width is measured based on knowledge about the height of the box which is computed by taking one point at the top of the box and one point at the bottom of the box as explained in the previous section.

In the first row, the measured width of the first image is 0.555m at the bottom and 0.581m at the top side. The two measurements are far from the actual width by around 1cm. Similarly, accurate measurements are obtained in the secondand third images of the first row. In the third image, the measurements obtained were still accurate despite the edges not being clear because of the dark color of the box. In the second row, the images were captured with a lower resolution camera so the measurements are not as accurate as the earlier ones. The actual width of the box is 0.342m; in the first image the measured width is 0.334m at the bottom and 0.367m at the top which is reasonably accurate. The measurement accuracy in the second image is accurate similar to the first image. In the third image, the measurement error is higher at the bottom side (3cm off the actual width). In general, the width measurements obtained by the proposed method has high accuracy because all the recorded errors are due to misdetection which could be improved by employing edge detection scheme in order to emphasize the edges of the object so it can be accurately detected.

Figure 19. Measuring the width of a box from different views.

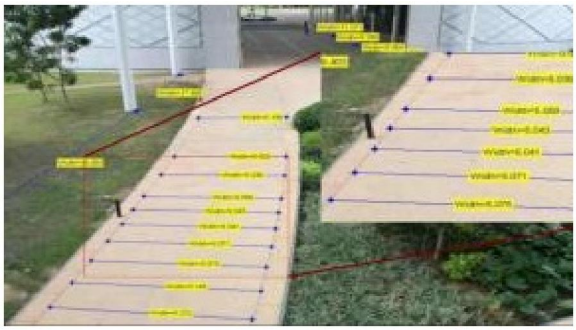
Figure 20. Measuring distance using triangulation algorithm.

In addition to computing width, DfT was employed for measuring distance between points in the image. Figure 20 shows measurement of the distance between two points in the image. The actual distance between the floor tiles shown in the image is 5.14m. This distance was measured at multiple points and the measurements are very accurate (the error is $\mp 1cm$).

Figure 21, shows distance measurement between two pillars in which images were captured using Canon 1000D camera installed at a height of 15.78m and with vertical angle 66 degrees. The actual distance between the pillars in the image is 22.50m. The measured distance between the first and second pillars is 22.75m; although the error is around 0.25m, this is considered to be very accurate because the points at which the height was measured are not carefully chosen and moreover the surface is not even. Similarly, the distance between the second and third pillars is 22.07m; this is also different from the actual distance by 0.43m. Figure 20 and Figure 21 illustrate that the triangulation method can be implemented for measuring distance by only

specifying the points of interest in the scene. This can be very helpful in aerial images becausethe height of the airborne vehicle and its vertical angle can easily be obtained from the navigation information.

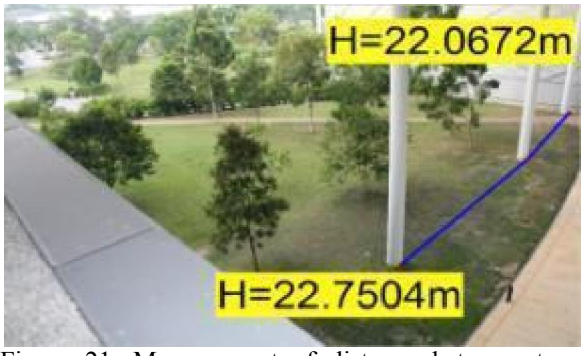

Figure 21. Measurement of distance between two points in the image using triangulation.

5.5 Uncertainty Analysis

In machine vision, there are various techniques that can be used for feature point selection; however all of these methods have a finite accuracy. In this section we analyze the effect of errors in image features selection. In addition, it considers the effect of errors in measuring the camera parameters such as height and vertical angle (known as calibration errors). This kind of analysis provides a sense of how uncertain the obtained measurement can be which is very important in machine vision applications.

Table 3 shows the error obtained due to 1% error in measuring camera height, camera angle or object location. For the object location, 1% error is relative to the image size. For example, if the image size is (640×480), 1% error means 6 pixels error in width and 5 pixels error in height. Table 3 shows error analysis for three types of error; if there is 1 degree error in measuring the pitch angle, it will yield to 1.75% error in computing the Y-axis coordinate, 1.75% error in computing the X-axis coordinate and 2.48% error in computing the depth of field. Similarly, 1% error in measuring the camera height will have effect on all the three coordinates by 1% for X and Y axis coordinates and 1.73% in the depth value. Errors in measuring a moving object location in the image is very common because existing object detection tools are not very accurate and thus 1% error is acceptable accuracy. 1% error in the width coordinate only affects the X-axis coordinate and the depth of field (Z) by 3.49% whereas 1% error in the height-coordinate will have effect on X, Y and Z coordinates by 3.49%, 3.49% and 4.94% of camera height respectively. This algorithm is very sensitive to error in the vertical direction compared to the horizontal one.

Table 3 Error analysis for the proposed method showing maximum possible error.

Input	Error	$\Delta\psi$	$\Delta\phi$	ΔX	ΔY	ΔZ
Pitch angle	1.00^0	1.0^0	0.0^0	1.75%	1.75%	2.48%
Camera height	1% of h	0.0^0	0.0^0	1.00%	1.00%	1.73%
x-axis	1% of W	0.0^0	2.0^0	3.49%	0.00%	3.49%
y-axis	1% of H	2.0^0	0.0^0	3.49%	3.49%	4.94%

6. Conclusion and Future Works

6.1Conclusion

This paper presented a method for computing the depth of field using the concept of triangulation. This method has many advantages over other depth estimation techniques. Firstly, it computes the depth from only one image captured from single 2D camera. In addition, this method does not require any prior knowledge about the scene content and it has short computational requirements compared to exiting techniques. This method utilizes basic camera setup information such as camera height, vertical angle and field of view for computing the 3D location of any point in the image using triangulation. The method is also extended for measuring the dimension of an object such as measuring the width and height as well as measuring the area of a selected object in the scene. In addition, the developed method can also be used for measuring actual distance between two multiple points in the image.

This method was tested with multiple images captured from different imaging devices. The accuracy of the estimated depth of field is influenced by the camera specifications; the higher the image resolution means the image element is smaller and hence it has smaller quantization error. DfT was tested by measuring the height of people with accuracy around 1cm. In addition, it has also been used for measuring the width and the height of a box from different view and a good accuracy was achieved as well. DfT was also implemented for measuring the distance between points in the image for both small and large distances; in both cases the measurement error is in range of centimeters which is considered to be small error for relatively large distances(greater than 15m). Finally, uncertainty analysis was presented for this method by identifying the possible sources of errors and studying how much they can affect the measured depth and geometry.

6.2Future Works

This algorithm can further be enhanced and extended for more applications. It can be extended to be used for 3D shape recovery by computing the depth for each point in the image. This requires finding ways of how to represent image points in such a way that DfT would be useful for computing their depth of field. In addition, the proposed algorithm can be enhanced by including an auto-calibration mechanism which can be used to automatically compute the camera height and vertical angle of a given image using some known landmarks in the image. Moreover, DfT can be implemented for measuring distances from aerial images as well as using it for robot navigation.

Corresponding Author:
Yasir Salih
Electrical Engineering Department
Faculty of Engineering and Islamic Architecture
Umm Al-Qura University
P.O. Box. 5555, 21955 Makkah, Saudi Arabia
E-mail: ysali@uqu.edu.sa

References

1. Barinova, O. et al., 2008. Fast automatic single-view 3-d reconstruction of urban scenes. In European Conference on Computer Vision. pp. 1–14.
2. Branca, A., Attolico, G. & Distante, A., 2002. Cast shadow removal in foreground segmentation. In International Conference on Pattern Recognition 16 (1). pp. 214–217.
3. Chan, P.P.K. et al., 2011. Depth estimatoin from a single image using defocus cues. In International Conferece on Machine Learning and Cybernetics. pp. 1732–1738.
4. Cornelis, N. et al., 2006. 3D city modeling using cognitive loops. In 3rd International Symposium on 3D Data Processing, Visualization and Transmission. pp. 9–16.
5. Criminisi, A., Reid, I. & Zisserman, A., 2000. Single View Metrology. International Journal of Computer Vision, 40(2), pp.123–148.
6. Das, A. et al., 2009. Improvised Filter Design for Depth Estimation from Single Monocular Images. In 3d International Conference on Pattern Recognition and Machine Intelligence. pp. 333–338.
7. Ewerth, R. & Schwalb, M., 2007. Using Depth Features to Retrieve Monocular Video Shots. In 6th ACM International Conference on Image and Video Retrieval (CVIR). pp. 210–217.
8. Futragoon, N., 2009. Enhanced Depth Estimation by Using Object Placement Relation. Computer Engineering, pp.1899–1904.
9. Geusebroek, J. & Smeulders, A.W.M., 2005. A Six-Stimulus Theory for Stochastic Texture. International Journal of Computer Vision, 62(1-2), pp.7–16.
10. Gould, S., Fulton, R. & Koller, D., 2009. Decomposing a scene into geometric and semantically consistent regions. In IEEE International Conference on Computer Vision. pp. 1–8.
11. Hedau, V., Hoiem, D. & Forsyth, D., 2009. Recovering the spatial layout of cluttered rooms. In 12th International Conference on Computer Vision. IEEE, pp. 1849–1856.
12. Hoiem, D., Efros, A. a. & Hebert, M., 2007. Recovering Surface Layout from an Image. International Journal of Computer Vision, 75(1), pp.151–172.
13. Hoiem, D. & Efros, A.A., 2009. Geometric Context from a Single Image. Processing.
14. Horry, Y., Tour Into the Picture : Using a Spidery Mesh Interface to Make Animation from a Single Image.
15. Jung, J. & Ho, Y., 2010. Depth map estimation from single-view image using object classification based on Bayesian learning. In 3D TV CON: The Ture Vision - Capture, Transmission and Display on 3D Video. pp. 1–4.
16. Kovács, L. & Szirányi, T., 2007. Focus area extraction by blind deconvolution for defining regions of interest. IEEE transactions on pattern analysis and machine intelligence, 29(6), pp.1080–5.
17. Kuo, T., Lo, Y. & Member, S., 2011. Depth Estimation from a Monocular View of the Outdoors. Current, pp.817–822.
18. Lalonde, J., Laganiere, R. & Martel, L., 2012. A single-view based obstacle detection for smart back-up camera systems. In IEEE International Conference on Computer Vision and Pattern Recognition Workshops. p. 2012.
19. Lee, D.-J., Merrell, P. & Wei, Z., 2010. Two-frame structure from motion using optical flow probability distributions for unmanned air vehicle obstacle avoidance. Machine Vision and Applications, 21(3), pp.229–240.
20. Lee, K.-Z., 2012. A simple calibaration approach to single view height estimation. In ninth Conference on Computer and Robot Vision. pp. 161–166.
21. Lee, Sang-yong et al., 2008. Enhanced Autofocus Algorithm Using Robust Focus Measure and Fuzzy Reasoning. IEEE Transactions on Circuits and Systems, 18(9), pp.1237–1246.
22. Lila, Y., Lursinsap, C. & Lipikorn, R., 2008. 3D shape recovery from single image by using

texture information. In International Conference on Control, Automation and Systems. IEEE, pp. 2801–2806.
23. Lin, C. & Chin, C., 2005. A novel architecture for converting single 2D image into 3D effect image. In The 9th International Workshop on Cellular Neural Networks and Their Applications. pp. 52–55.
24. Liu, B., Gould, S. & Koller, D., 2010. Single image depth estimation from predicted semantic labels. In IEEE International Conference on Computer Vision and Pattern Recognition. pp. 1253–1260.
25. Malik, A.S. & Choi, T.S., 2008. A novel algorithm for estimation of depth map using image focus for 3D shape recovery in the presence of noise. Pattern Recognition, 41(7), pp.2200–2225.
26. Mendonça, P.R.S. & Kaucic, R., 2008. Single View Metrology : A Practical Example. In IEEE Workshop on Applications of Computer Vision. pp. 1–8.
27. Nagahara, H., Ichikawa, A. & Yachida, M., 2008. Depth estimation from the color drift of a route panorama. In IEEE/RSJ International Conference on Intellignet Robots and Systems. pp. 22–26.
28. Park, S.W., Heo, J. & Savvides, M., 2008. 3D Face Reconstruction from a Single 2D Face Image. In IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–8.
29. Peng, K. et al., 2010. Single View Metrology Along Orthogonal Directions. In 2010 20th International Conference on Pattern Recognition. IEEE, pp. 1658–1661.
30. Pribyl, B., Zemcik, P. & Republic, C., 2011. Simple Single View Scene Calibration. Lecture Notes in Computer Science (Advances Concepts for Intelligent Vision Systems), 6915, pp.748–759.
31. Ribeiro, E. & Hancock, E.R., 1999. Improved pose estimation for texture planes using multiple vanishing points. In International Conference on Image Processing. pp. 148–152.
32. Rother, D., Patwardhan, K. a. & Sapiro, G., 2007. What Can Casual Walkers Tell Us About A 3D Scene? In 2007 IEEE 11th International Conference on Computer Vision. IEEE, pp. 1–8.
33. Saxena, A., Chung, S.H. & Ng, A., 2008. 3-D Depth Reconstruction from a Single Still Image. International Journal on Computer Vision, (76), pp.53–69.
34. Saxena, A., Sun, M. & Ng, A.Y., 2009. Make3D: learning 3D scene structure from a single still image. IEEE transactions on pattern analysis and machine intelligence, 31(5), pp.824–40.
35. Scharstein, D., Szeliski, R. & Zabih, R., 2001. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001). IEEEComput. Soc, pp. 131–140.
36. Shimodaira, H., 2006. A shape-from-shading method of polyhedral objects using prior information. IEEE transactions on pattern analysis and machine intelligence, 28(4), pp.612–24.
37. Suzuki, M.T., Yaginuma, Y. & Kodama, H., 2009. A texture energy measurement technique for 3D volumetric data. In IEEE International Conference on Systems, Man and Cybernetics. IEEE, pp. 3779–3785.
38. Torralba, A. & Oliva, A., 2002. Depth estimation from image structure. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(2), pp.1226–1238.
39. Wang, G., Wu, Y. & Hu, Z., 2002. A novel approach for single view based plane metrology. In International Conference on Pattern Recognition. pp. 556–559.
40. Wang, S. et al., 2008. Shape from Shading based on Maximum Entropy. In Computer Science and Information Technology, 2008. ICCSIT '08. International Conference on. pp. 287–289.
41. White, R. & Forsyth, D. a., 2006. Combining Cues: Shape from Shading and Texture. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Vol. 2, pp. 1809–1816.
42. Xu, L.-Q., Kandabaso, J.L. & Pardas, M., 2005. Shadow removal with blob-based morphological reconstruction for error correction. In ICASSP, Acoustics, Speech, and Signal Processing.
43. Zhang, R. et al., 1999. Shape from Shading : A Survey. IEEE Transaction on Pattern Analysis and Machine Intelligence, 21(8), pp.690–706.

9/11/2013