

## An Image Based Approach to Compute Object Distance

Ashfaqr Rahman\*

Department of Computer Science, American International University Bangladesh  
Dhaka 1213, Bangladesh

Abdus Salam, Mahfuzul Islam, and Partha Sarker†

Image Processing Group, Department of Computer Science, American International University Bangladesh  
Dhaka 1213, Bangladesh  
www.aiub.edu

Received: 26-04-2008

Revised: 09-08-2008

### Abstract

Computing object distance using image processing is an important research area in the field of computer vision and robot navigation applications. In this paper we have proposed a new method to compute the distance of an object using a single image. According to our observation there exists a relationship between the physical distance of an object and its pixel height. We exploit this relationship to train a system that finds a mapping between an object's pixel height and physical distance. This mapping is then used to find the physical distance of test objects from the pixel height in the image. Experimental results demonstrate the capability of our proposed technique by estimating physical distance with accuracy as high as 98.76%.

*Keywords:* depth detection, object tracking, image processing, artificial intelligence.

### 1. Introduction

Introduction of artificial intelligence<sup>1</sup> has prompted researchers to think about various applications including automated robot navigation<sup>2</sup>, computer vision etc. Several challenges in these areas include object detection<sup>3</sup>, obstacle avoidance, and location finding etc. Image processing techniques<sup>4</sup> are widely used in recent times in the field of computer vision<sup>5</sup> and robot navigation to address the abovementioned problems. Although other hardware based approaches using sonar, laser etc. is also used to find obstacles; image processing techniques are so far offering maximum accuracy. While dealing with obstacle avoidance or

object recognition, an important task in many approaches is to find how far the object is from the camera. In this paper we propose an image based technique to find the distance of objects.

Image based distance computation techniques can be classified into two groups – (i) Stereo vision based techniques and (ii) Mono vision based approach. Stereo vision based methods<sup>6</sup> use two cameras to find the depth information. It uses two well-positioned cameras to find the depth map of the image. It takes two images at the same time and then finds the disparity map using a very complex method. Although more accurate, this technique is time consuming as multiple images of the same object need to be processed. Moreover

---

\* Dr. Ashfaqr Rahman, Assistant Professor, Department of Computer Science, American International University Bangladesh (AIUB). *Mail:* Campus 4, 55/B, Road 21, Kemal Ataturk Avenue, Banani, Dhaka 1213, Bangladesh. *Ph:* +8801713408082 (M) *Email:* ashfaqr@aiub.edu

† *Mail:* Image processing group, Department of Computer Science, American International University Bangladesh (AIUB), Campus 4, 55/B, Road 21, Kemal Ataturk Avenue, Banani, Dhaka 1213, Bangladesh.

A. Rahman et al.

deployment of such a technique is expensive as it requires two cameras.

Mono vision based approach on the other hand are comparatively less expensive as it requires only one camera. Different mono vision based techniques are there in the literature<sup>7-10</sup>. In Ref. 7 object depth is computed based on the amount of image resizing in proportion to the camera movement. This technique needs to be supplied the object size and the camera parameters. Moreover it requires the camera movement to take multiple images to compile the decision thus making the process computationally expensive.

The mono vision based approach in Ref. 8 includes two steps: (i) It calculates an interpolation function based on the height and the horizontal angle of the camera, and (ii) It then uses this function to calculate the distance of the object from the camera. The limitation of this method is the dependency on primitive height and horizontal angle. There are also some other algorithms that can find depth of an object using single camera. But all of them have some drawbacks. Some of these techniques<sup>9</sup> use image sequences to find the distance that makes it time consuming and thus limiting its application in real time. We also find some approaches<sup>10</sup> that uses single camera to find object distance but these use some auxiliary devices such as laser, LEDs etc.

A careful scrutiny of the existing literature reveals that vision based approaches in general need auxiliary devices, in addition to camera and raise the cost. Other vision based approaches need the processing of multiple images thus resulting in higher time consumption. We are thus motivated to develop mono vision based approach that is realizable in real time without sacrificing accuracy. Our proposed technique utilizes the relationship between pixel height of an object and the physical distance. The relationship is learned in terms of known pixel heights and physical distances. Distances of test objects are found by mapping the pixel height to physical distance. As shown in the experimental results our proposed technique computes physical distance with high accuracy and is less sensitive to noise.

The remainder of this paper is organized as follows: Section 2 describes our proposed technique. We present some experimental results in Section 3. Finally Section 4 concludes the paper.

## 2. Our Proposed Method

We are motivated to compute object distance using a technique that sailors commonly use to find how far they are from the shore. The sailors normally use the projected height of the light house to find the approximated distance of the shore. We use a similar philosophy to find the distance of object from the projected height (pixel height) of the object.

Computing the actual distance of the object is decomposed into computing the depth and lateral distance of the object (Fig 1). From our observation the depth of the object is related to the pixel height of the object. The lateral distance at a particular depth, however, is almost independent of the object height. We thus use an alternate technique to find a mapping between actual and pixel lateral distances. Once both depth and lateral distance is computed we can compute the actual object distance as:

$$\text{Actual distance} = \sqrt{[\text{Depth}]^2 + [\text{Lateral distance}]^2} \quad (1)$$

Both depth and lateral distance computation techniques are elaborated in the following sections.

### 2.1. Mapping Object Depth from Pixel Height

A general observation (Fig 2) is that if an object goes far from the camera, the object has a reduced size in the image. As the object moves closer to the camera the object becomes comparatively bigger. This observations lead to the fact that the object depth has a direct relationship with its pixel height in the image. For this reason we are motivated to place an object at different positions and compute the depth and pixel height of the object. The pixel height of the object remains same at a particular depth even if we change the lateral distance. Table 1 presents the relationship between depth and pixel height for a typical rectangular object of height 30 cm. A pictorial representation of this table in Fig 3 reveals that a polynomial curve can approximate the trend very well. For the rectangular object the best fitting second order polynomial equation is as follows:

$$\delta = 0.0033 \times \rho^2 - 1.9918 \times \rho + 372.83 \quad (2)$$

where  $\delta$  is the object depth and  $\rho$  is the pixel height.

With this above mapping (2) in hand we can place the object at an arbitrary position within the viewpoint of the camera. From the captured image we can find the pixel height (using any edge detection technique in Ref.

9) and use the map in (2) to find the depth of the object. We can generalize the abovementioned concept for any arbitrary shape of objects.

## 2.2. Mapping Object Depth from Pixel Height

For any types of object it is also possible to get the lateral distance from the camera center by using the object's coordinate. From our observation the size of an object at a particular depth is almost independent of the lateral distance. Thus the ratio between lateral physical distance and lateral pixel distance from the camera center of an object is almost constant at a particular depth (Fig 4). It varies only a little. For this reason we have taken mean of the ratios for a particular depth by placing the object at different lateral distances from the camera center. This ratio however varies over the depth of the object. Table 2 presents the relationship between object depth and physical–pixel lateral distance ratio for a typical rectangular object of height 30 cm. Fig 5 pictorially represents this table. It reveals that a polynomial curve can approximate the trend very well. For the rectangular object the best fitting second order polynomial equation is as follows:

$$\text{Ratio} = 0.0004 \times \delta^2 - 0.1469 \times \delta + 18.119 \quad (3)$$

where  $\delta$  is the object depth.

With this above mapping (3) in hand we can place the object at an arbitrary position within the viewpoint of the camera. From the captured image we can find the pixel height and use the map in (2) to find the depth of the object. By putting the depth in (3) we can compute the ratio. With the pixel lateral distance in hand (computed using image processing<sup>11</sup>) we can use this ratio to find the physical lateral distance of the object. We can generalize the abovementioned concept for any arbitrary shape of objects.

The basic platform of our proposed technique relies on learning the two relationships in (2) and (3). The organization of our proposed technique is presented in Fig 6. During the learning stage we learn the two relationships for the objects under consideration. During testing phase we use this learned mappings to find the actual object depth.

## 3. Experimental Results

We have conducted a diverse set of experiments to verify the correctness of our proposed technique. In this section we present our experiments. We start this

section with the experimental platform and tools that we have used that is followed by the results.

In our experiment we have used objects of four shapes: (i) Rectangular, (ii) Triangular, (iii) Cylindrical, and (iv) Circular. The first three shapes were of 30 cm height and circular ball was of diameter 6.5 cm. From our observations we found that all of the first three objects have almost equal pixel heights in the image. We were thus motivated to use the same polynomial learning for these three types of objects. In order to distinguish between object types in an overlapping situation they were painted using different colors. The objects were painted with non-reflective colors so that light is not reflected. We used uniform (white color) background while capturing the images. The images were taken at 640×480 resolution. We considered sufficient lighting condition in the environment. While capturing images we assumed that the rectangular, triangular and cylindrical objects are orthogonal to the floor. Please note that the proposed technique can also be used under different lighting conditions and non uniform background as long as the object height is computed accurately.

Implementation of our proposed technique requires few image processing tools. For noise elimination and efficient edge detection images are filtered using Kuwahara filter<sup>12</sup>. Images are then converted to gray scale to apply Sobel operator<sup>13</sup> for edge detection. The edges are used to identify the objects. These tools are used during the testing phase. The complete implementation of the project is available in Ref. 14.

The proposed system starts by learning a map between the features of the captured image and object depth and lateral distance of an object. During learning images are taken by placing the objects at different positions (within the viewpoint of the camera). We then measured the physical depth and lateral distance (in centimeters) of the object. The pixel height and lateral distance of the objects in the image are then obtained using MS Paint (Fig 7). The readings in different locations are then used to generate the two polynomial mappings. We used Microsoft Excel to generate the polynomials. The best fitting second order polynomials as obtained during the experiments for depth and physical–pixel lateral distance ratio is presented in (2) and (3).

We have developed a C# application to automate the proposed technique. It takes an image as input and tells

A. Rahman et al.

the depth and lateral distance of the object using the equations generated in learning phase. After an image is taken a Kuwahara filter is applied to reduce the noise and sharpen the edges of objects. Then the image is converted to grayscale and sobel edge detection algorithm is applied to find the edges of the objects. These edges are used to decide on the shape of the objects (circular or rectangular). The shape information is necessary as the pixel height computation method is different for rectangular and circular objects. The computed pixel height and pixel lateral distances are used with the learned polynomials to get the physical depth and lateral distance of the objects. Finally we use the computed depth and lateral distances to calculate the actual physical distances of the objects using (1).

We have tested the effectiveness of the proposed technique by placing the objects at different locations, and orientations. Fig 8 and Fig 9 presents two such scenarios. In Fig 8 there is no overlapping between the objects. Table 3 provides a comparison between the experimental results and actual physical distances. It can be observed that the error is on an average 1.185%. On an average this is very high accuracy. Fig 9 presents an image with overlapping objects. Table 4 provides a comparison between the experimental results and actual physical distances. It can be observed that the error is on an average 1.315% that is very high accuracy. It can be observed in both cases that the circular object possess maximum error. This is because the circular object is the smallest object. Thus the same error has higher impact on the circular object than the other three objects. The main reason of error for circular objects is the shadow of the object that sometimes is considered as part of the object during height calculation.

From the experimental data, we can say that our proposed method is very easy and less error prone to find the actual distance of the object. The accuracy of the results depends on how well the system has been developed and learned. The system can be deployed in scenarios where it is necessary to know the accurate position of the object with the environment learned in advance. In this method there is no depth limit but it will fail to find distances for those objects which appear partially in the image.

#### 4. Conclusions

In this paper, we have introduced a new image based technique to find the depth and lateral distance of

different types of objects. Our proposed technique computes depth and lateral distance of objects with high accuracy. This method has some advantages such as it is very simple, accurate, requirement of no additional devices, use of a single camera, low computational complexity etc. Our proposed technique, however, is limited to only few basic shapes like rectangular, cylindrical, triangular and circular objects are dealt with in this method and all of them are rigid. Currently we are working to make this method more generic.

#### References

1. S. Russell and P. Norvig, "Artificial Intelligence A Modern Approach," Pearson Education, ISBN: 81-297-0041-7, 2003.
2. Optical Flow Based Robot Navigation, [http://people.csail.mit.edu/lpk/mars/temizer\\_2001/Optical\\_Flow/](http://people.csail.mit.edu/lpk/mars/temizer_2001/Optical_Flow/), Last accessed in April 2008.
3. P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 511, vol. 1, 2001.
4. Y. Q. Shi and H. Sun, "Image and video compression for multimedia engineering," CRC Press, 2000.
5. D. A. Forsyth and J. Ponce, "Computer Vision: A Modern Approach," Prentice Hall, 2002.
6. J. Carnicelli, <http://www.alexandria.nu/ai/blog/entry.asp?E=32>, Last accessed in July 2007.
7. N. Yamaguti, Sh. Oe, and K. Terada, "A Method of Distance Measurement by Using Monocular Camera," The 36th SICE Annual Conference, 1997.
8. M. Mirzabaki and A. Aghagolzadeh, "Introducing a New Method for Depth Detection by Camera using Lagrange Interpolation," The Second Iranian Conference on Machine Vision, Image Processing & Applications, 2003.
9. Y. L. Murphey et. al., "Depth Finder: A Real-time Depth Detection System for Aided Driving," IEEE Intelligent Vehicles Symposium, 2000.
10. M. Takatsuka and et al., "Low-Cost Interactive Active Monocular Range Finder," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1999.
11. R. C. Gonzalez, R. E. Woods, "Digital Image Processing" – Second Edition, Prentice Hall.
12. Kuwahara filter, <http://www.incx.nec.co.jp/imap-vision/library/wouter/kuwahara.html>, Last accessed in July 2007.
13. Sobel Edge Detector, <http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>, Last accessed in October 2008.
14. A. Salam, M. Islam, and P. Sarker, [http://partha.bengalit.net/paper/monovision\\_aiub](http://partha.bengalit.net/paper/monovision_aiub), Last accessed in July 2007.

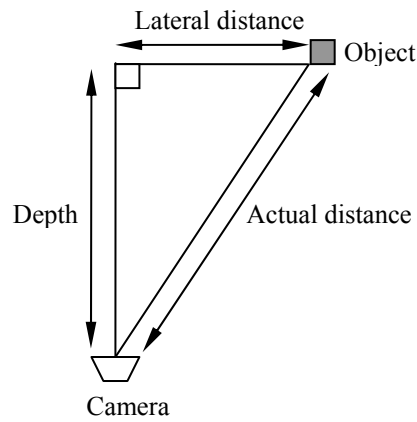


Fig 1: Top view of visualizing object distance. Computation of actual distance is decomposed into computing depth and lateral distance.

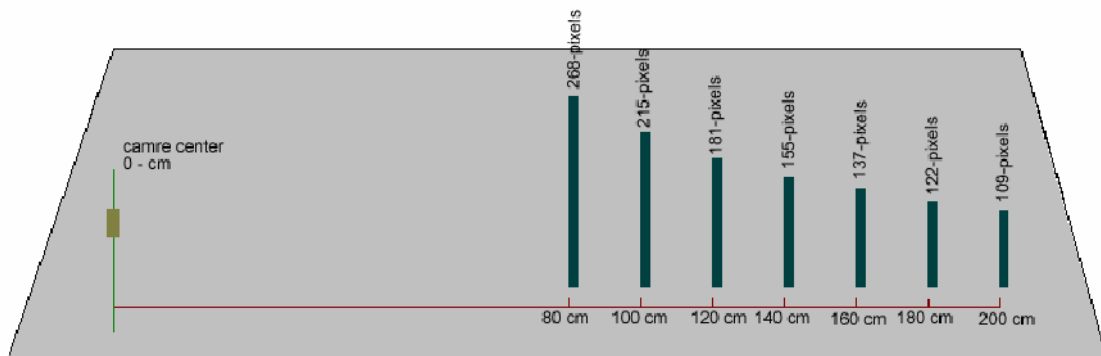


Fig 2: Comparison of same object's depth and pixel height.

Table 1: Pixel height of an object at different physical distances

Pixel height	307	268	241	215	197	181	167	155	146	137	128	122	115	109
Depth (cm)	70	80	90	100	110	120	130	140	150	160	170	180	190	200

A. Rahman et al.

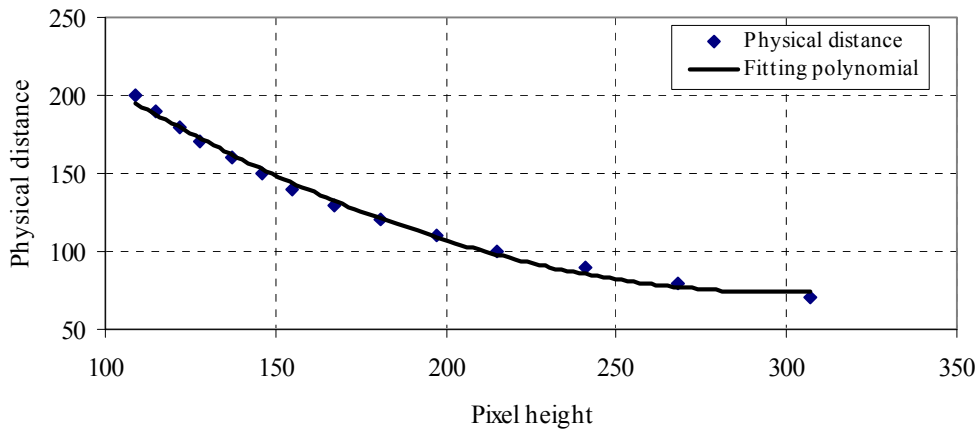


Fig 3: Fitting polynomial representing the relationship between object depth and pixel height.

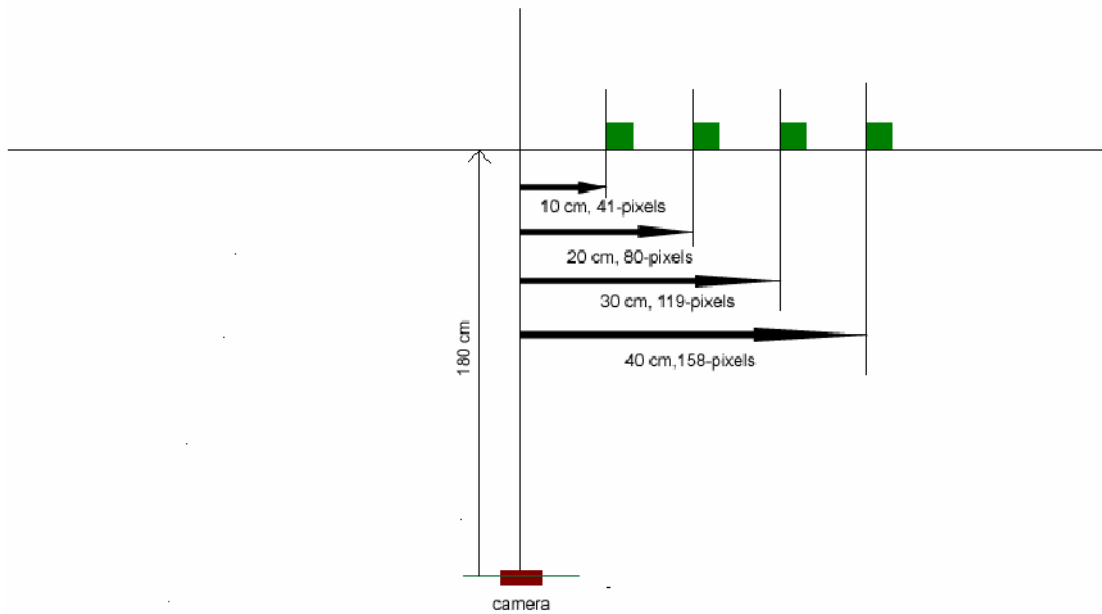


Fig 4: Comparison of lateral physical and pixel distance at object depth 180 cm. The ratio is approximately equal to four.

Table 2: Physical–pixel lateral distance ratio of an object at different depth

Depth (cm)	70	80	90	100	110	120	130	140	150	160	170	180	190	200
Ratio	10	8.7	7.8	7	6.4	5.9	5.4	5	4.67	4.4	4.17	3.97	3.8	3.65

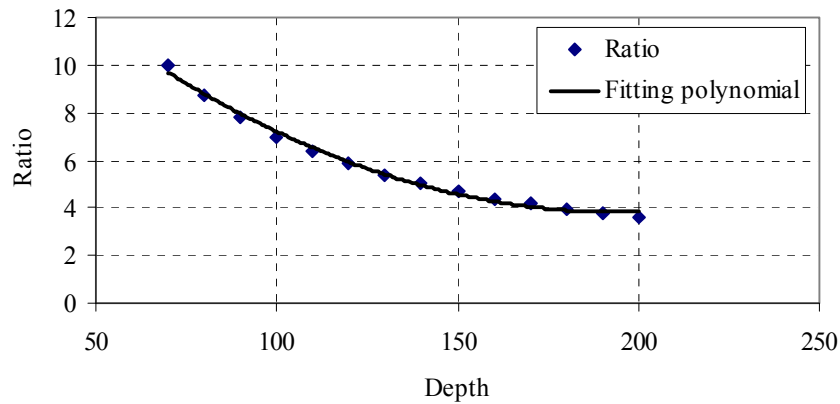
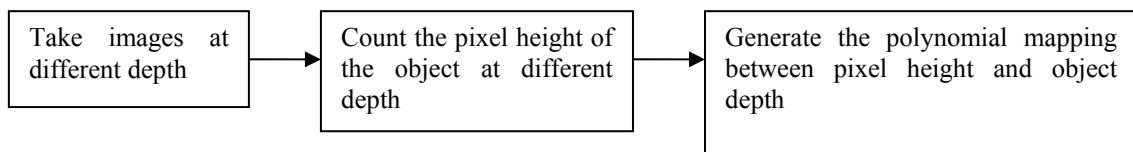
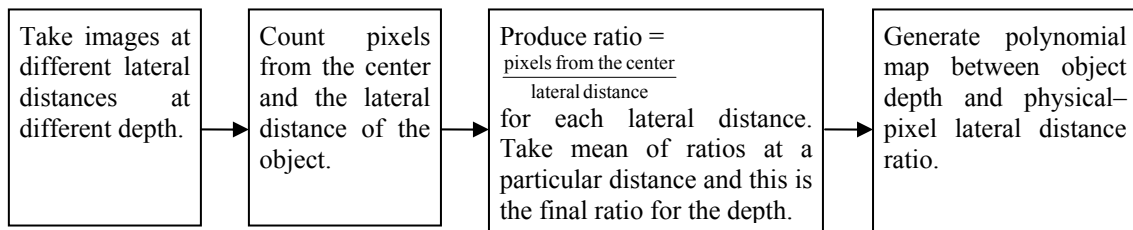


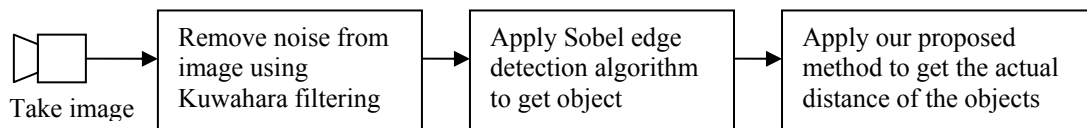
Fig 5: Fitting polynomial representing the relationship between physical–pixel lateral distance ratio and depth.



(a) Learning a map between object depth and pixel height



(b) Learning a map between object depth and physical–pixel lateral distance ratio



(c) Applying the learned maps to compute object distance in real environments.

Fig 6: Flow diagram of the proposed technique.

A. Rahman et al.

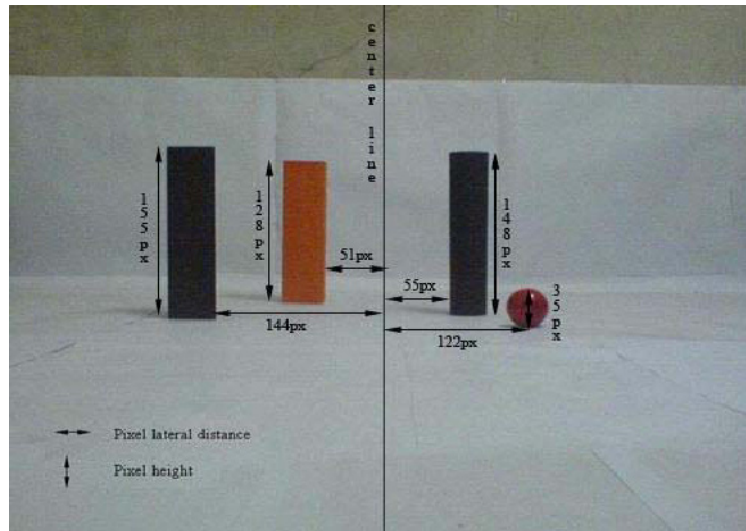


Fig 7: Computation of pixel height and pixel lateral distance of a set of objects.

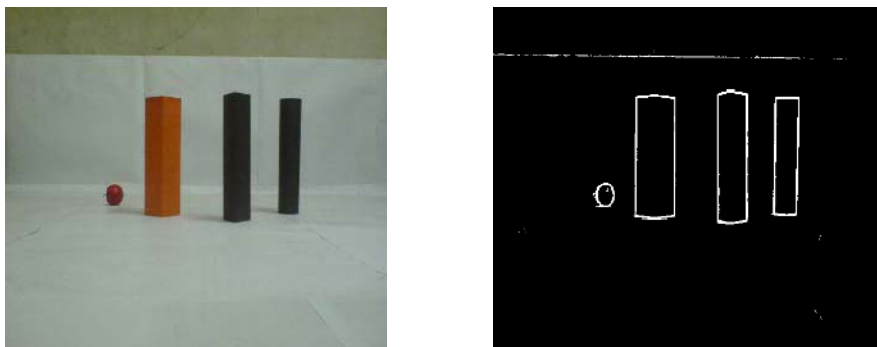


Fig 8: Objects at depth 120, 130, 140, 170 cm and 8, 5.5, 25, 30 cm lateral distance.

Table 3: A comparison of actual and computed object distance for Fig 8.

	Black triangular object	Orange rectangular object	Black cylindrical object	Circular object
Actual depth (cm)	120	130	140	170
Calculated depth (cm)	121.73	131.62	138.76	167.56
Actual lateral distance (cm)	8	5.5	25	30
Calculated lateral distance (cm)	7.13	4.9	26.48	29.35
Actual physical distance (cm)	120.27	130.12	142.21	172.63
Calculated physical distance (cm)	121.94	131.71	141.26	170.11
Error in percentage (%)	1.39	1.22	0.67	1.46



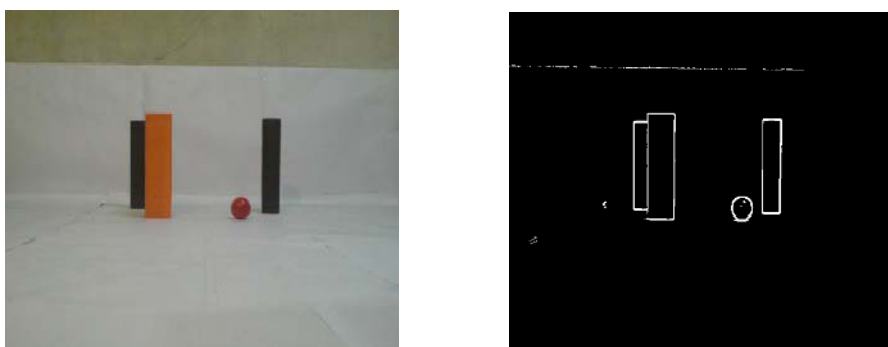


Fig 9: Objects at depth 151, 170, 180, 150 cm and 12, 20.5, 22, 12 cm lateral distance.

Table 4: A comparison of actual and computed object distance for Fig 9.

	Orange rectangular object	Black triangular object	Black cylindrical object	Circular object
Actual depth (cm)	151	170	180	150
Calculated depth (cm)	152.30	168.29	177.44	152.88
Actual lateral distance (cm)	12	20.5	22	12
Calculated lateral distance (cm)	10.15	20.31	19.15	11.97
Actual physical distance (cm)	151.48	171.23	181.34	150.48
Calculated physical distance (cm)	152.64	169.51	178.47	153.35
Error in percentage (%)	0.77	1.00	1.58	1.91