Machine Vision for General Cameras for Quality Testing and Dimension Calculations

Ashwath Narayan Murali

***Abstract*—This paper looks into an economical way to bring machine vision to smartphones and basic camera modules without any major setup using basic image processing techniques. This technique can be used to measure the dimensions of random objects as well as be used to detect particular objects in an image. A single camera module can be used for both purposes i.e. measure objects whose image distance is known and measure dimensions of objects whose image distance needs to be calculated, making the process efficient, economical and portable. We look into a new area where machine vision can aid the general population and be affordable to small scale industries as well.**

***Index Terms*—Machine vision, image processing, quality assurance, dimension calculation, portability.**

I. Introduction

The paper wishes to bring to light the process of using machine vision to accurately measure the dimensions of objects to observe quality and also be used to measure dimensions of objects in an image using various image processing techniques. The implementation of this can help improve the efficiency of quality assurance and also make high quality products in terms of dimensions. Using this method, any general camera can be used to perform machine vision and the set up time is negligible compared to current systems.

There are two types of machine vision that are required, one where the distance of the object from the lens is known and the focal length of the lens is known along with parameters such as resolution of the image [1]. The other case is where the distance of the object from the camera lens is unknown but other parameters such as resolution, focal length etc. are known. This paper uses a single camera module to perform both types of image processing.

II. Concept

For the first type of machine vision, i.e. when the distance of the object from the camera lens and other parameter are known, the dimensions of the object in 2D can be measured by using the following equation.

1) Object Height(pixels)
2) Sensor Height(mm)
3) Focal Length(mm)
4) Image Height(pixels)
5) Distance To Object(mm)

$$\frac{ObjectHeight_{pixels} \times SensorHeight_{mm}}{FocalLength_{mm} \times ImageHeight_{pixels} \times DistToObject_{mm}}$$

The Above equation will give the user the real height of the object in mm units. Based on the units, the user can obtain results in different units.

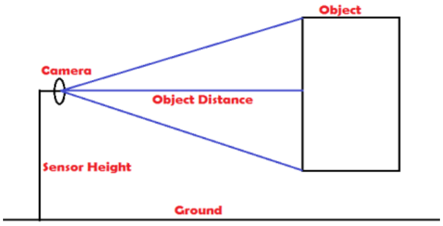

Fig. 1. Basic requirement for machine vision.

This is the basics of machine vision using optics is displayed in Fig. 1. Now, when processing the image, line by line processing is done to improve accuracy. But to ensure that processing time is reduced, there should be a balance between the resolutions of the image and processing power should be maintained. If the resolution is high, the processing time is more but accuracy is high and vice versa [2].

$$Process_{time} \sim Resolution_{image}$$

The lighting should be adjusted in such a way such as to avoid shadows. This is crucial as shadows reduce the clarity of the image and extrapolates its dimensions and requires a more complex algorithm to process the image and also reduces the accuracy as rounding off is required if shadows are involved. Also, using clear images makes the processing easier and scheduling [3] them in a processor is done more efficiently for a clear image.

The second case is when the user wants to measure the dimensions of a random object in an image. This brings in certain problems and requires a different approach. The primary requirement for this is to know the distance between the object and the camera. Using autofocus, the camera can get a rough value of the distance of the object from the camera. This single feature also gives us the focal length, aperture etc. of the camera at that point. Once the image is obtained and the parameters are known, the size of an object in the image can be calculated by the number of pixels between the object. That gives us the object distance in pixels. To get the distance in metric scale, we need to know the resolution of the image and the resolution of the screen in which it is displayed. Both these details are readily available. The key point about the resolution is the need to know the DPI (Dots per Inch) or PPI (Pixels per inch) of the image and

Manuscript received May 10, 2015; revised July 23, 2015.

Ashwath Narayan Murali is with the SASTRA University, Thanjavur, India (e-mail: 115014019@sastra.edu).

DOI: 10.7763/IJMO.2015.V5.477

the screen. Using these concepts and basics of optics, the system can calculate the dimensions of any object in an image. To detect particular objects in an image, the particular object to be detected is compared with the image by comparing its RGB value. This is a quick and effective way to spot similar objects in an image. A database storing the image to be detected is required to compare with. Scaling is used to detect the object in an image as the scaling may vary.

III. Working

To calculate the dimensions of objects from an image, certain parameters are required. These parameter values need to be accurate independently. Once implemented, the camera can be used to calculate any object dimension as it is versatile. To effectively calculate the dimensions of an object from an image, we need the following parameters: To calculate the object height, a scan line algorithm is used to calculate the distance between two selection points on the image. This value may vary, but it will not affect the object height to image height ratio. Both are in pixels and the ratio will be constant independent of the screen used to view the image. This requires line by line scanning of the image starting at the first selection point to the second.

The complex part is to retrieve the object distance from the camera lens [4]. This can be done using various methods. These methods are elaborated as follows.

1) Using a reference: In this method, a reference image of known dimensions is used to measure the distance of the object, this method is done for known objects and select objects. This process uses generalization as its main principle, and hence is prone to errors for non-standard objects.
2) Moving the camera: This method uses multiple reference points using the same camera. The image is taken at different locations and hence the multiple reference points are present to gauge the dimension of the objects. The distance between the reference points can be calculated using an accelerometer.
3) Depth from Focus/Defocus: This is the problem of estimating the 3D surface of a scene from a set if 2 or more images of that scene. The images are obtained by changing the camera parameters (typically the focal setting or the image place axial position), and taken from the same point of view.
4) Using an IR beam to get the distance: This method is the most reliable method. An IR beam is used to get the distance of the object. The object to be focused is selected by the autofocus, this is a very accurate method but is expensive as not all smartphones come with an IR autofocus.
5) Using the autofocus: The autofocus of the camera itself can be used to get the distance between the focused object and the camera. This is not as accurate the IR beam, but can be complimented using any of the above methods.

The next step is to get the height of the sensor. This is the height of the camera from the ground. This information is required for accuracy of the result. The height of the sensor when it comes to basic photography is around 5 feet. But it varies significantly based on the user. The height can be calculated using the accelerometer present in smart phones or reference images can be taken to determine the height accurately.

IV. Image Height Calculation

The image height depends on the camera resolution and the PPI (Pixels Per Inch) of the screen in which the image is viewed for processing. To calculate the PPI of the screen where the image is obtained, we need the following details.

1) Diagonal of the screen/image in inches (usually for the screen).
2) Diagonal of the screen/image in pixels.

The diagonal of the screen/image in pixels can be calculated using the following formula:

$$d_p = \sqrt{}(w_p^2 + h_p^2)$$

Now, the PPI is given by the formula:

$$PPI = \frac{d_p}{d_i}$$

where d_p is the diagonal in pixels and d_i is the diagonal in inches.

The focal length f of the camera will be specified by the manufacturer and the resolution of the camera will also be specified.

A. Case 1: Using Reference Image

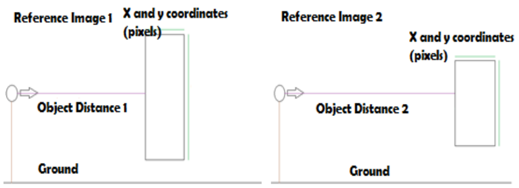

Fig. 2. Using Reference of image 1 to get dimension of image 2.

In this method, we use the camera to take images at different distances. The ratio of the object scaling based on pixels being occupied in the image can be used to roughly determine the distance of the object from the camera. From Fig. 2.

A=Reference one
B=Reference two
$X1$=Scaling of X coordinate in image A (Pixels)
$Y1$=Scaling of Y coordinate in image A (Pixels)
$X2$=Scaling of X coordinate in image B (Pixels)
$Y2$=Scaling of Y coordinate in image B (Pixels)
K= Constant

$$\left(\frac{X1}{Y1}\right)K = \frac{X2}{Y2}$$

Use a standard image as a reference image, we can find the distance of an arbitrary image by scaling it till LHS and RHS are equal. Given the distance. This is an accurate method provided the initial calculations are accurate.

B. *Case 2: Moving the Camera*

This method is strictly for calculating distances when the object is stationary. Moving the camera stimulates a stereo pair. This makes it possible to get different snapshots of the image at different positions. Using trigonometry and basic math, the distance can be calculated. Fig. 3 and Fig. 4 show the different possibilities under this case.

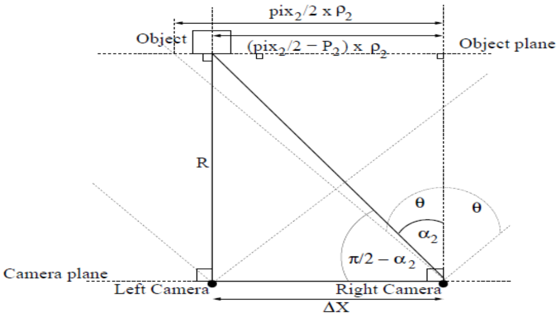

Fig. 3. When the object is in front of the camera.

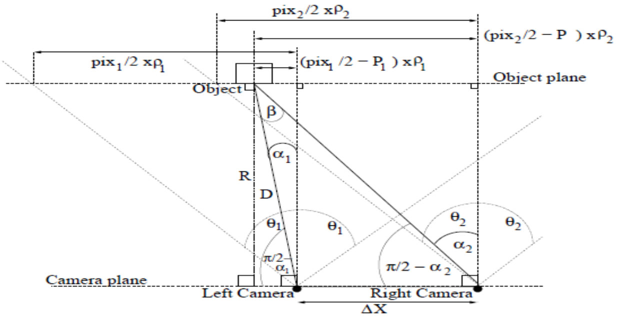

Fig. 4. Arbitrary position of the camera.

Based on the location of the object from the camera, we can get multiple equations to calculate the distance.

When the object is in front of the left position/camera:

$$R = \tan\left(\frac{\pi}{2} - \alpha_2\right) \times \Delta X$$

where, *R*= Distance from camera.

$$\alpha_2 = tan^{-1}\left(\frac{\frac{pix_2}{2} - P2}{\frac{pix_2}{2}} \times tan\Theta_2\right)$$

When the object is in front of right camera/position:

$$R = \tan\left(\frac{\pi}{2} - \alpha_1\right) \times \Delta X$$

where,

$$\alpha_1 = tan^{-1}\left(\frac{P1 - \frac{pix_1}{2}}{\frac{pix_1}{2}} \times tan\Theta_1\right)$$

$$R = \frac{\sin\left(\frac{\pi}{2} - \alpha_1\right) \times \sin\left(\frac{\pi}{2} - \alpha_2\right) \times \Delta X}{\left(\sin\left(\alpha_2 - \alpha_1\right)\right)}$$

The above equation is for the object at the left. For right:

$$R = \frac{\sin\left(\frac{\pi}{2} - \alpha_1\right) \times \sin\left(\frac{\pi}{2} - \alpha_2\right) \times \Delta X}{\left(\sin\left(\alpha_1 - \alpha_2\right)\right)}$$

where *R* is the distance, α1 is the angle with the first camera or the 1st image and α2 is the angle with the first camera or the 2nd image,

C. *Case 3: Depth from Focus/Defocus*

In this method, the camera focus or the camera parameters are changed to get different reference images to calculate the distance of an object from the camera. This method is used when the camera used are real aperture cameras. Real aperture cameras have a short depth of field, resulting in images which appear focused only on a small 3D slice of the scene. Using the thin lens law

$$\frac{1}{f} = \frac{1}{v} + \frac{1}{u}$$

where *f* is the focal length, *u* is the distance between the lens place and the plane in focus in the scene and *v* is the distance between the lens plane and the image plane. Using this method, a very simple image is obtained, which is easy to process and read and produces an opaque Lambertian (i.e. with constant bidirectional reflectance distribution function.) In this case the intensity *I*(*y*) at a pixel $y\epsilon Z^2$ of the CCD surface can be described by:

$$I(y) = \int (h_u(y, x, s(x)) r(x) dx$$

where the kernel *h* depends on the surface *s* and the optical settings *u* and *x* is proportional to the square of R for a fixed surface *s*(*x*) = d, the kernel h is function of the difference *y*-*x* i.e. integral above becomes the convolution.

$$I(y) = (h_{u',d} \times r)(y)$$

In simple terms the kernel *h* determines the amount of blurring that affects a specific are of the surface in the scene. This helps to identify objects in the image and also help in calculating the distance by the peak value of the Gaussian surface plotted against the kernel h. Fig. 5 gives the parameters behind a simple test case.

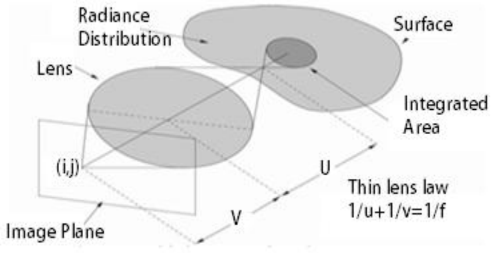

Fig. 5. Image detection by depth from focus/defocus.

D. Case 4: Using the IR Distance Sensor

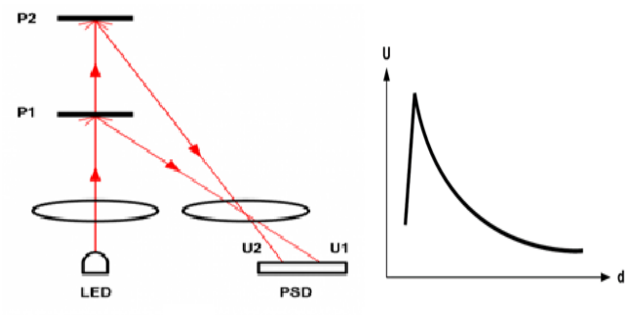

Fig. 6. Using IR to get the distance using PSD corresponding graph for response.

The IR sensor which provides accurate details about the distance of the object from the camera works on the principle of reflection of light rays. The IR rays are reflected from the object and the reflected light is absorbed by the sensor, its voltage value is used to establish the distance between the object and the camera. This method can be used to calculate the dimensions for moving objects. [2] Fig. 6 shows the working of the IR sensor to calculate distance.

The IR distance sensor contains a LED and a PSD (Position Sensible Detector)

As the distance of the object increases, the voltage of the reflected light is reduced, this is usually a linear graph and can give the distance accurately. The general equation to calculate the distance is given by

$$D = \left(\frac{1}{P \times ADC + Q} \right) - K$$

where,

D=Distance in cm.
K= Corrective constant.
ADC=Digitized value of voltage.
P=Linear member
Q=Free member

To improve the range of the IR sensor, a lens can be placed in front of the LED, but it will reduce the accuracy as the distance increases. Fig. 7 illustrates this using a graph.

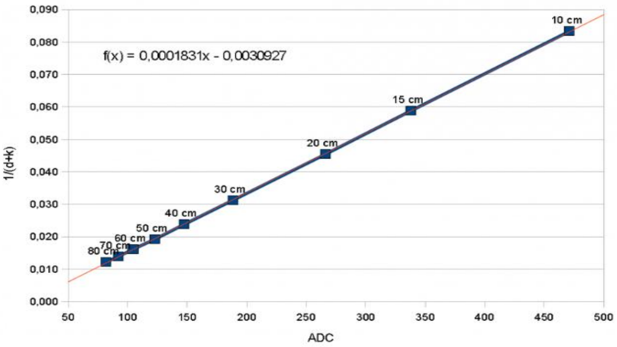

Fig. 7. Sample results using IR sensor for range.

E. Case 5: Using Autofocus Parameter

Using the camera autofocus feature is a rough estimate to calculate the distance of the object from the distance. Any camera with autofocus holds the distance value of any object that the camera is focusing on. This distance value is approximate and not highly accurate. This value is stored in the memory and can be fetched to get the distance. The accuracy depends on the autofocus algorithm used. The drawback is that using autofocus render the distance to many objects in an image as infinity with respect to the camera. This will not give any result in the dimension calculation and object detection.

V. Limitations

The major limitation of the above methods is the accuracy. The distance between an object and the camera is very difficult to get accurate measurements. Using multiple methods to capture the image can help improve the accuracy [5], but will increase the processing time as well.

Any object which is very far away from the camera cannot be detected accurately as this will be at infinity with respect to the lens. Hence, it can be used only on nearby objects.

The accuracy of the dimensions and object detection is dependent on the camera specs. The better the camera resolution and focusing capabilities, better the accuracy.

VI. Practical Applications

The use of machine vision to calculate dimensions is a simple and efficient way to calculate object dimensions in all aspects. This makes it easy and automated. Machine vision incorporated in smartphone cameras make it portable and affordable to the general public and also can help people in the field of construction and industrial engineering. This method reduces the cost for setting up quality assurance for small scale industries as the components required are economical to purchase and use. The further use of machine vision can be implemented for the following:

1) Detect common objects in an image by using the data library on the smart phone.
2) Detect numbers, logos, email ID's etc based on scan line algorithms.
3) Be positioned at a particular point to scan objects for defects and be used as a quality assurance device[6].
4) Make scanning of barcodes, QR codes etc simple and efficient without any third party application.
5) Connect to the internet to compare image with the internet to aid in online e-commerce etc.

These applications of machine vision using our design principles and concepts make it affordable and useful to a larger market. Using multiple images [7], [8], using the different methods, high accuracy rates can be achieved.

VII. Conclusion

Using the above methods, a simple camera module present in a smart phone or on an independent module or barebones can be used to perform quality assurance and dimensional calculations. When complemented with a repository of images, it can be used to recognize patterns such as numbers, objects etc. from an image.

The above applications make machine vision versatile as well as economical and reduce the set up costs for quality control.

REFERENCES

[1] D. Petkovic, "The need for accuracy verification of machine vision algorithms and systems," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 1989, pp. 439-440.

[2] M. Arsalan and A. Aziz, "Low-cost machine vision system for dimension measurement of fast moving conveyor products," in *Proc. 2012 International Conference on Open Source Systems and Technologies (ICOSST),* 2012, pp. 22-27.

[3] A. R. Nolan, B. Everding, and W. Wee, "Scheduling of low level computer vision algorithms on networks of heterogeneous machines," in *Proc. Computer Architectures for Machine Perception,* 1995, pp. 352-358.

[4] F. Espinosa, C. Gordillo, R. Jimenez, and O. Aviles, "Dynamic traffic light controller using machine vision and optimization algorithms," in *Proc. 2012 Workshop on Engineering Applications (WEA),* 2012, pp. 1-6.

[5] C. Vigneswaran, M. Madhu, and R. Rajamani, "Inspection and error analysis of Geneva gear on machine vision system using Sherlock™ and VB 6.0 Algorithm," in *Proc. 2012 International Conference on Machine Vision and Image Processing (MVIP),* 2012, pp. 193-196.

[6] H. Sako, "Recognition strategies in machine vision applications," in *Proc. Machine Vision and Image Processing Conference,* 2007, pp. 3.

[7] H. Y. Sun, C. J. Sun, and Y. H. Liao, "The detection system for pharmaceutical bottle-packaging constructed by machine vision technology," in *Proc. 2013 Third International Conference on Intelligent System Design and Engineering Applications (ISDEA),* 2013, pp.1423-1425.

[8] K. Iyshwerya, B. Janani, S. Krithika, and T. Manikandan, "Defect detection algorithm for high speed inspection in machine vision," in *Proc. 2013 IEEE International Conference on Smart Structures and Systems (ICSSS),* 2013, pp.103-107.

Ashwath Narayan Murali was born in Bangalore, India, in 1993. He received his bachelor of technology (information and communication technologyt) from SASTRA University, Thanjavur, India, in 2015.

He has published three papers in the *International Journal of Emerging Technology and Advanced Engineering* (IJETAE) in vol. 4, no. 6, in the field of computer science. He is currently researching in algorithms and mobile computing.

Mr. Murali has won awards for his research work at National Institute of Technology Trichy, National Institute of Technology Calicut, Indian Institute of Space Science and Technology Trivandrum.