# Prediction of XYZ coordinates from an image using mono camera

**Muslikhin[1], D Irmawati[1], F Arifin[1], A Nasuha[1], N Hasanah[2] and Y Indrihapsari[3]**

[1]Electronics Education Department, Universitas Negeri Yogyakarta, Yogyakarta 55281,Indonesia
[2]Informatics Education Department, Universitas Negeri Yogyakarta, Yogyakarta 55281, Indonesia
[3]Information Management Department, National Taiwan University of Science and Technology, Taiwan

**Abstract.** Estimating the position of a homogeneous object from an image for XY position is quite simple because it has the same dimensions XY. However, determining the XYZ position requires a unique approach. Generally, for estimating 3D position, stereo camera or expensive cameras are used with complicated computer vision algorithms. In this paper, we classify the position of an object using a mono camera. The image is divided into 3185 classes and five layers as a machine learning algorithm references. The k-nearest neighbors (kNN) approach usually is to find the closest point of the centroids to the closest class. Thus, this approach can be used as a three-axis prediction method that can afford the best performance solution.

## 1. Introduction

Prediction of a coordinate in robotics has become an essential part, especially reading XYZ coordinates for mobile and manipulator robots. These robots usually have the task to identify the position of particular objects. In the identifying position process, there are two popular application; hand to eye and eye in hand. The hand to eye is relatively easy because the position of the camera relative to the object is always fixed. Vice versa, the eye in hand has a relative position between the object and the camera. In other words, because the camera is attached to the end effector, each movement of the robot joint will be affected in a position relative to an object [1].

The idea of predicting the position of XYZ objects was developed from XY. To predict objects with XY, using a projection ratio pattern can determine the location of objects based on the capture area. The challenge with adding the Z-axis occurs because we play in three dimensions while the image is only two dimensions. That approach has been carried out with stereo cameras. The stereo cameras can overcome the XYZ position, but in terms of construction, it is quite wasteful [2] whereas coordinate prediction using monocular cameras has the opportunity to be more compact [3]. With its compact construction, it allows the application to monitor more swiftly and not to disturb movement.

In short, in order to position prediction process can be accurate the Z position needs to be divided into layers. Each layer will be divided into small classes. The class will be labels with their respective ordinate. The next task is to find the centroid of an object and match it with the machine learning approach: k-nearest neighbors (kNN) [3][4]. Therefore, the focus of this paper is to find the best performance for XYZ predictions using the kNN.

KNN is an essential part of the XYZ position prediction process [5]. Exceptionally, the kNN approach is needed to be proofing before being applied for an eye in hand manipulator. This paper is organized as follows: the monocular camera described in section 2 continues with kNN in section 3. Developing a prediction of XYZ position using kNN is in section 4. We describe the experimental result in section 5. Definitely, conclusions are drawn in section 6.

## 2. Mono Camera

The mono camera is different from the monocular camera even though in number is same (one unit). Monocular has a structure like the human eye with circular projections, while the mono camera with rectangular projections [6] [7]. However, both have similarities in the pinhole principle, as does the webcam, which is a mono camera. The webcam structure has a pinhole like the following.

### *2.1. Pin hole camera*

The pinhole model is a mathematical basis used to develop 3D point relationships in object space as (P) with coordinates (x1, x2, and x3) and projections in the image field (Q) with coordinates (y1, y2). This projection is only an estimate by eliminating the lens on the camera [5]. Pinhole cameras have an essential role as a camera calibration modality, which does not include geometric distortion, optical aberration, or blur. As an illustration, Fig. 1 interprets the pinhole camera model.

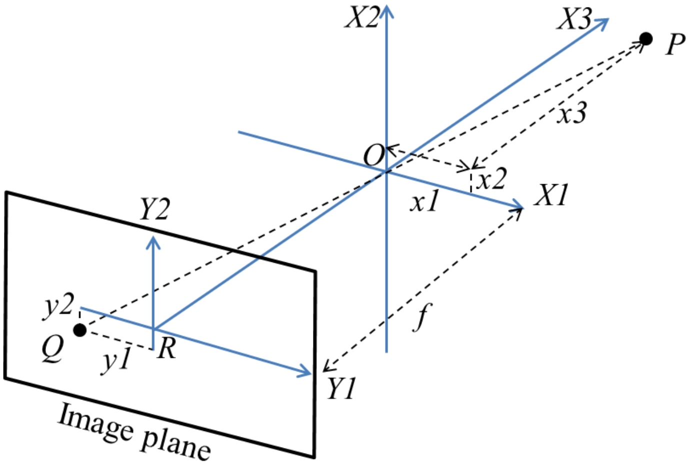

**Figure 1.** Pin hole camera model

The position of O is a camera pinhole, while viewed towards the camera, there are three axes X1, X2, and X3. Image plane describes as a projected image and is at a focal length (f). Point R is the intersection between the optical axis and the image plane, in other words, as the center of the image. The association between two coordinate systems (objects and image fields) can be calculated using the similarity of triangles as described in Eq. 1 and 2.

$$y_1 = \frac{-f\, x_1}{x_3} \tag{1}$$

$$y_2 = \frac{-f\, x_2}{x_3} \tag{2}$$

### *2.2. Boundary extraction*

Boundary extraction is part of mathematical morphology which is very often used in image processing. The basic concept of the boundary extraction is based on the process of erosion of binary imagery by structuring elements —the boundary of set A, denoted by β(A). The boundary results obtained from eroding A by structuring element (B) are continued by looking for a binary difference between the sets A and B [7].

$$\beta(A) = A - (A \ominus B) \tag{3}$$

Figure 2 illustrates the boundary extraction mechanism. This shows simple binary objects, structuring elements, and results using Eq. 3. Although structuring element in Fig. 2.b is the most commonly used. For instance, using structuring element B will produce a boundary between 2 and 3 pixels.

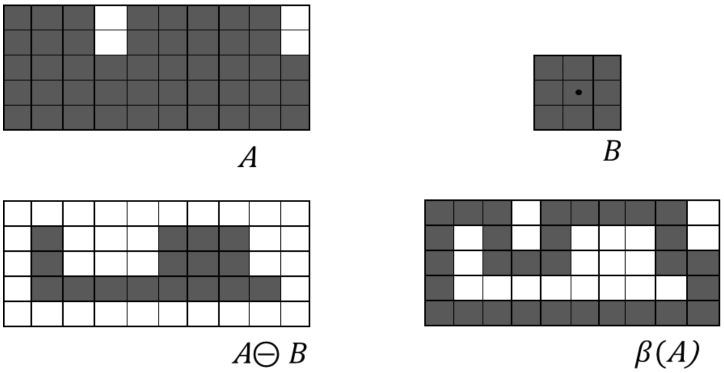

**Figure 2.** a) Set binary image, b) structuring element, c) eroded by d) boundary

### *2.3. Hole filling*

The definition of a hole can be interpreted as a region surrounded by pixels different from the background. In this section, develop algorithms based on set dilation, complementation, and intersections to fill holes in the image. Spouse an image with each boundary enclosing a background area (i.e. a hole). Then each hole is given a point 0s (black), the goal is to fill all black holes with logic 1s (white) [8].

We start by forming an array, $X_0$ from 0s (with the same array size). Then, the following procedure for filling all holes with 1 is as Eq. 4. The B is the symmetric structuring element in Fig. 2b. The algorithm terminates at iteration step k if $X_k = X_{k-1}$. The set $X_k$ then contains all the filled holes [9]. The set union of $X_k$ and $A$ contains all the filled holes and their boundaries.

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1,2,3,\ldots \tag{4}$$

The dilation process will fill the entire holes if left unchecked. However, we can limit it in the desired region. Fig. 3b is the rest of the dilation process that is limited, in case a black dot will disappear from the middle of the circle as well as proof of Eq. 4.

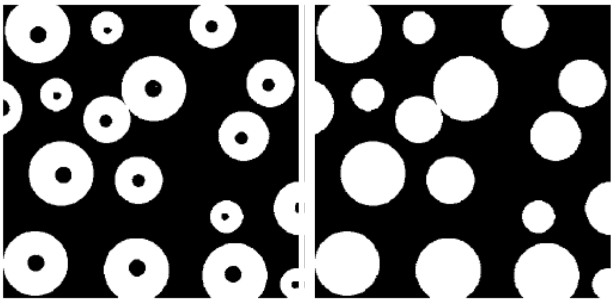

**Figure 3.** a) Binary image white dot inside of the regions, b) result of filling holes

## 3. k-Nearest Neighbors (kNN)

kNN can be used for classification problems and predictive regression. However, in the industry, it tends to be widely used for classification [10]. The classification process in kNN comes from distance calculation. These calculations can use various methods such as Euclidean, Manhattan, Cebycev, Mahalanobis, City block, Minkowski, Cosine, Hamming, Jaccard, and Spearman distance. The Euclidean distance (Ei) was chosen to be used in this paper and obtained from Pythagoras calculations. The length of the X-axis is attained from the difference x1 and the end axis x2. The Y width axis is reached from the initial y1-axis and y2-axis, while the high Z-axis is obtained from z1 and z2. The Ei between the target position and the reference point is illustrated in Eq.5.

$$E_i = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \qquad (5)$$

The given a set of n points and distance functions in capture areas, kNN search allows us to find the closest point k to X, Y, and Z. The search technique and kNN-based algorithm are widely used as a benchmark for learning rules. The value of k indicates the number of lines between set points with class points. This means that the more the number of k positions will be well performed.

## 4. Predicting of XYZ Position using kNN

The dataset predicts the position of an image in this paper taken from the height of the camera 642 mm, 592 mm, 542 mm, 492 mm, and 442 mm. The image is divided into small areas called classes so that there are 15925 datasets in total. The dataset will be a reference point for calculating kNN with k=1, 2, and 4.

### *2.4. Simulation setting*

In this paper, the simulation is done using MATLAB 2017 software. The mono camera is working HD Pro Webcam C920 with 640x480 in resolution. Dataset values for estimators are around 15925 datasets, as shown in Table 1. The prediction of XYZ coordinate for an object was made in limited area 510x415x200 mm. Computers used Intel Core i5 CPU @ 3.0 GHz (6 CPUs), 16 GB of RAM, with Windows 10 Pro 64 bit operating system.

### *2.5. The kNN recognizes the XYZ position using mono camera*

To predict the position of the object using machine learning: kNN is through the following path. First, take a photo with a certain height. Second, convert image to grayscale and then to a binary image. Third, do the process of boundary extraction and filling the hole. Fourth, after the release of the centroid and the perimeter label, these two data are used as input to find the new point value to the nearest class in the dataset. Finally, display the closest data according to the data set and k numbers as depicted in Fig. 4.

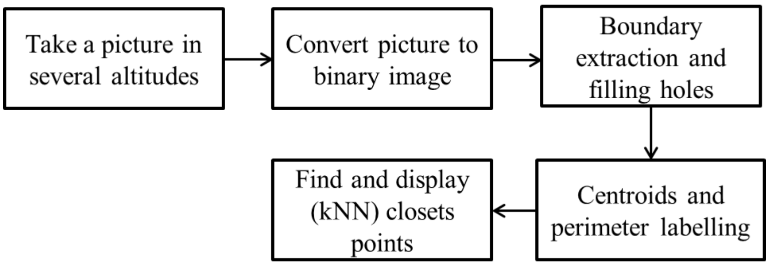
Take a picture in several altitudes
Convert picture to binary image
Boundary extraction and filling holes
Centroids and perimeter labelling
Find and display (kNN) closets points

**Figure 4.** kNN predicts the XYZ position is taken with a mono camera

kNN compares data centroids with three-variable datasets (XYZ). Prediction accuracy performance is strongly influenced by the number of k, the huge k the accuracy will better. The points in the picture show the midpoint of each class. The class area in this trial is 100 pixels square. Thus the farthest distance of centroids to objects is a maximum of a quarter of the class area or 25 pixels. Let us look at Fig.5 showing (*) as the centroid of an object, while a black circle is k=1 with a point in the middle, which is the closest class result based on kNN calculations. The same way applies to 2nd, third targets, etc.

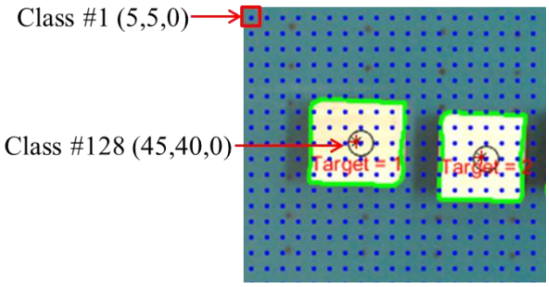

**Figure 5.** Correlation kNN dataset toward classes and object centroids

If we use k=1, the proximity of predictions depends very much on the accuracy of the class so we cannot find the value of the slices of both. Thus, for k=2, it is quite challenging to find out slices. The ideal number of k is at least 3 for each field, horizontal and vertical. However, the number of k that is too much k> 3 will not have a significant effect on improving predictive performance. This description only applies to this case, where the distribution of classes is structured. Let us consider the k and value illustrations 2 and 4 in Figure 6, visible centroids of the object "X" and the prediction of the center position of the object (+) in a circle with a fixed black dot (.).

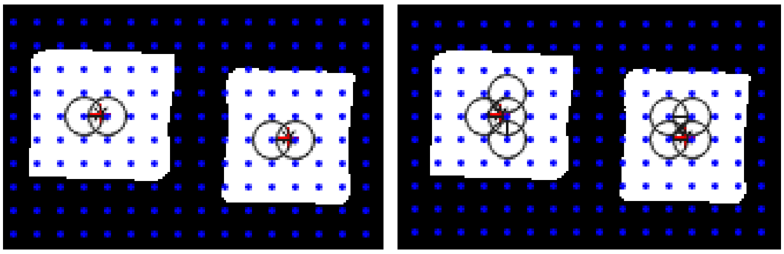

**Figure 6.** Comparison between number of k=2 and k=4

## 5. Experiment Result

The simulation results are seen in the level of performance in predicting the position using the kNN. The value of k=1 (lowest) was chosen to analyze performance. A number of 15925 datasets are expected to be able to cover the lack of k. The results of the experiment with k=1 data are collected as in Table 1, and we show proportionally for this paper.

**Table 1.** XYZ comparison after kNN applied in an image

| image (pixels) | | | coordinate (mm) | | | coefficients | | Verifications (mm) | | | deviation ± (mm) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Xi | Yi | Zi (per.) | Xr | Yr | Zr | XY | Z | Xv | Yv | Zv | X | Y | Z |
| 0 | 0 | *211.12* | 0 | 0 | 0 | 0.79 | 0.16 | 0.00 | 0.00 | 0.28 | 0.00 | 0.00 | -0.28 |
| 240 | 0 | *234.28* | 172.50 | 0 | 50 | 0.71 | 0.18 | 172.50 | 0.00 | 50.22 | 0.00 | 0.00 | -0.22 |
| 480 | 0 | *263.47* | 307 | 0 | 100 | 0.63 | 0.20 | 307.01 | 0.00 | 99.39 | 0.02 | 0.00 | 0.60 |
| 0 | 320 | *309.01* | 0 | 180 | 150 | 0.56 | 0.23 | 0.00 | 180.37 | 149.45 | 0.00 | -0.37 | 0.54 |
| 240 | 320 | *340.71* | 116 | 155 | 200 | 0.48 | 0.26 | 116.01 | 154.68 | 199.29 | -0.01 | 0.31 | 0.70 |
| 480 | 320 | 211.12 | 382 | 255 | 0 | 0.79 | 0.16 | 382.04 | 254.69 | 0.28 | -0.04 | 0.30 | -0.28 |
| 0 | 640 | 340.71 | 0 | 310 | 200 | 0.48 | 0.26 | 0.00 | 309.36 | 199.29 | 0.00 | 0.63 | ***0.70*** |
| 240 | 640 | 340.71 | 116 | 310 | 200 | 0.48 | 0.26 | 116.01 | 309.36 | 199.29 | -0.01 | 0.63 | 0.70 |
| 480 | 640 | 309.01 | 270 | 360 | 150 | 0.56 | 0.23 | 270.56 | 360.74 | 149.45 | ***-0.56*** | ***-0.74*** | 0.54 |

It can be seen in Table 1 that the deviations for XYZ position are the largest at 0.56 mm, 0.74 mm, and 0.7 mm, respectively. In the image column, there should Xi and Yi only which are 2D forms of the image plane, but Zi (perimeter) is added to the column which is the result of boundary to find the perimeter (number of pixels) value. There are five perimeter areas (italics) which correlate with the height of the object. Next Xr, Yr, and Zr are the coordinate position of objects in mm. Obtaining two images and coordinate column data, we can calculate coefficients which can then be used to verify and detect several deviations. The other three data labeled Xv, Yv, and Zv, are the results of verification. According to functions of kNN is to predict, and then the regression function plays a role in this matter. Determining the XY position depends on Z, which will affect the perimeter (see Table 1 coefficient column) based on the existing regression equation. Regression values for finding a 2D image to a 3D position are obtained from regression in an example, Eq. 6-7. Next, we can replace the coefficients according to the perimeter value, as shown in Table 1.

$$Y_{(z=50)} = 0.71878\, X_{(z=50)} \tag{6}$$

$$Z_{i\ (per.)} = (234.2\ \times 0.18022) + 8 \tag{7}$$

A little confusing when reading the XYZ position of the experimental image results is 2D. Therefore we visualize with varying viewpoints to clarify the coordinates of an object in MATLAB.

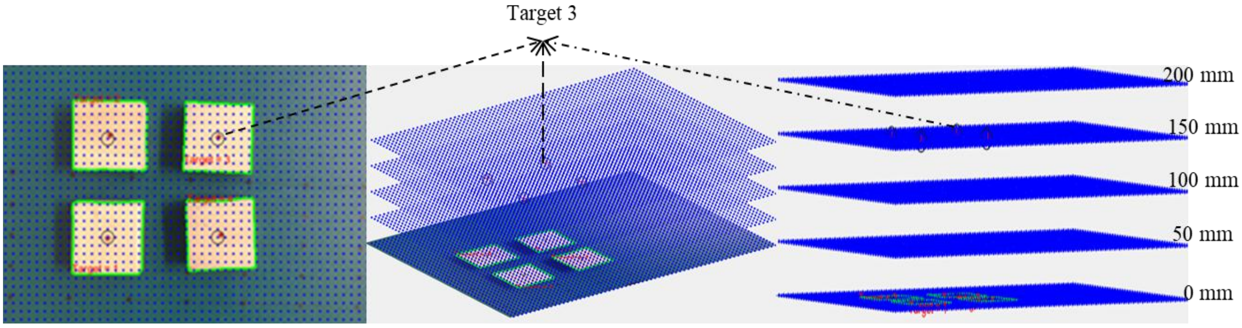

**Figure 7.** The detected object using kNN a) top view, b) diagonal side view (azimuth = 30°, elevation = 20°), c) diagonal side view (azimuth = 30°, elevation = 2°)

Look at Fig. 7. 2D illusion on the image as if it was at an altitude of 0 mm. The azimuth angle is given, and the elevation value was increased by 20°, which glance better but is still difficult to interpret. Now the elevation angle is lowered to 2°; it can be construed clearly like Fig. 7.c. that the object is at an altitude of 150 mm form the while the camera position at an altitude of 642 mm.

## 6. Conclusion

This paper takes pictures of an object using a camera with a different Z position — wooden beams of uniform size as specimens. The experiment was completed using MATLAB. The XYZ position prediction results can be completed by kNN (k=1), and the closest distance is obtained by calculating the Euclidean distance. The position of the object is modeled to visualize and verify position predictions. The prediction results show that XYZ analysis performs well, even with k=1 because it is overcome by many datasets. In terms of performance, the system shows acceptable deviations with a value of <1 mm.

## 7. References

[1] Aashna S et al 2015 Monocular camera based object recognition and 3d localization for robotic grasping Int. Conf. on Signal Processing Computing and Control (Allahabad) vol– (Uttar Pradesh: India) p 225

[2] Iwata T *et al* 2017 Improvement of object position estimation accuracy using hyperbola and ellipse *Proc. Int. Symp. on Antennas and Propagation (Phuket)* vol -- (Phuket: Thailand) p 1

[3] Kim J *et al* 2018 Mod: multi-camera based local position estimation for moving objects detection *IEEE International Conference on Big Data and Smart Computing (Shanghai)* vol 1 (Shanghai: China) p 642

[4] Song S *et al* 2016 Multiple objects positioning and identification method based on magnetic localization system *IEEE Transactions on Magnetics* 52 1

[5] Xuanmin L *et al* 2016 An improved dynamic prediction fingerprint localization algorithm based on knn *Sixth International Conference on Instrumentation & Measurement, Computer, Communication and Control (Harbin)* vol 1 (Harbin: China) p289

[6] Han J *et al* 2016 Vehicle distance estimation using a mono-camera for fcw/aeb systems Int. Journal of Automotive Technology 17 483

[7] Kumar M S S *et al* 2013 Face distance estimation from a monocular camera *IEEE Int. Conf. on Image Processing (Melbourne)* vol 1 (Victoria: Australia) p 3532

[8] Rajwant K and Sukhpreet K 2013 Object extraction and boundary tracing algorithms for digital image processing: comparative analysis: a review. *Int. Journal of Advanced Research in Computer Science and Software Engineering* 3 263

[9] Satou T *et al* 2012 Simple three-dimensional measurement method for vehicle using a monocular camera *The 1st IEEE Global Conference on Consumer Electronics (Tokyo)* vol 2 (Tokyo: Japan), p 427

[10] Yadav R K *et al* 2019 Trusted k nearest bayesian estimation for indoor positioning system *IEEE Access* 7 51484