

# Monocular vision and calculation of regular three-dimensional target pose based on Otsu and Haar-feature AdaBoost classifier

Yuanhong Li<sup>1,2,3</sup>, Hongjun Wang<sup>1,2\*</sup>, Weiliang Zhou<sup>1</sup>, Zehao Xue<sup>1</sup>

(1. Department of Engineering, South China Agricultural University, Guangzhou 510642, China;

2. Southern Key Laboratory of Agricultural Equipment Machinery, South China Agricultural University, Guangzhou 510642, China;

3. Department of Biological and Agricultural Engineering, Texas A&M University, College Station, TX, 77843, USA)

**Abstract:** Using machine vision to identify and sort scattered regular targets is an urgent problem to be solved in automated production lines. This study proposed a three-dimensional (3D) recognition method combining monocular vision and machine learning algorithms. According to the color characteristics of the targets, to convert the original color picture into  $YC_bC_r$  mode and use the 2D Otsu algorithm to perform gray level image segmentation on the  $C_b$  channel. Then the Haar-feature training was carried out. The comparison of feature training and Haar method for Hough transform showed that the recognized time of Haar-feature AdaBoost trainer reached 31.00 ms, while its false recognized rate was 3.91%. The strong classifier was formed by weight combination, and the Hough contour transformation algorithm was set to correct the normal vector between plane coordinate and camera coordinate system. The monocular vision system ensured that the field of camera view had not obstructed while the dots were being struck. It was measured and calculated angles between targets and the horizontal plane which coordinate points of the identified plane feature. The testing results were compared with the Otsu and AdaBoost trainer where the prediction and training set have an error of no more than 0.25 mm. Its correct rate can reach 95%. It shows that the Otsu and Haar-feature based on AdaBoost algorithm is feasible within a certain error ranges and meet the engineering requirements for solving the poses of automated regular three-dimensional targets.

**Keywords:** Otsu, Haar-feature, AdaBoost, 3D position, target pose, monocular vision, error analysis

**DOI:** 10.25165/j.ijabe.20201305.5013

**Citation:** Li Y H, Wang H J, Zhou W L, Xue Z H. Monocular vision and calculation of regular three-dimensional target pose based on Otsu and Haar-feature AdaBoost classifier. *Int J Agric & Biol Eng*, 2020; 13(5): 171–180.

## 1 Introduction

Visual algorithm technology includes object shape recognition, velocity sensing, distance recognition and image pattern recognition technology<sup>[1,2]</sup>. Among those applications, the technology of 3D recognition is most prominent, especially in the actual industrialization pipeline. The detection of small targets has always been the hotspots and difficulties in the research field of image processing and its methods are more diverse<sup>[3-5]</sup>. With the development of industrial automation technology and the detection of targets on the production and logistics transmission line, due to the fast running speed, many types of targets and large distance changes, current machine vision recognition technology seems to be powerless<sup>[6]</sup>, especially for the complex background environment, the diversity of color targets and the complicated segmentation algorithm between the small targets of the pipeline. Although the target can be accurately identified, running speed is slow resulting in low efficiency of the robot, which directly affects the pipeline and the production efficiency<sup>[7]</sup>.

In order to improve the recognition rate of targets on the pipeline, some of simple, high real-time performance and fast

running speed algorithms become the hotspots of target image exploration. The Otsu method is a classical threshold segmentation algorithm proposed by Otsu in 1979<sup>[8,9]</sup>. It has attracted widespread attention due to its good segmentation effect, wide application range, and simplicity and effectiveness. This paper proposes to use Otsu to segment pipeline targets with its high segmentation accuracy and strong adaptability<sup>[10,11]</sup>. This improved Otsu search threshold mode can quickly calculate the target segmentation threshold and theoretically optimize the iterative efficiency. The algorithm designs a new threshold recognition function to replace the traditional two-dimensional segmentation threshold by the threshold of two one-dimensional Otsu methods. To ensure maximum integrity within the classification, a minimum dispersion within the class is introduced<sup>[12]</sup>. This method not only considers the grayscale variation and distribution information of pixels but also considers the domain space information around the pixels<sup>[13]</sup>. In general, the search ability of Otsu in the global scope is very powerful. In the actual working scene, after the work-piece is visually segmented by the monocular, it is necessary to solve the posture of the plane of role target to determine the relative normal vector on the camera plane.

The AdaBoost algorithm is iterative. Its core idea is to train different weak learners for the same training set, and then combine the weak learners to form and create a strong learner<sup>[14,15]</sup>. The AdaBoost algorithm based on Haar-feature training has the advantages of high prediction accuracy and strong generalization ability. It has wide use in neural networks and machine learning feature classification<sup>[16]</sup>. This study used it to classify the pose of the regular three-dimensional target in the actual working

**Received date:** 2019-03-03 **Accepted date:** 2020-06-08

**Biographies:** Yuanhong Li, PhD candidate, research interest: agricultural engineering and machine vision, Email: liyuanhong@stu.scau.edu.cn; Weiliang Zhou, MS candidate, research interest: agricultural mechanization engineering, Email: 799733220@qq.com; Zehao Xue, MS candidate, research interest: agricultural mechanization engineering, Email: m13602725637@163.com.

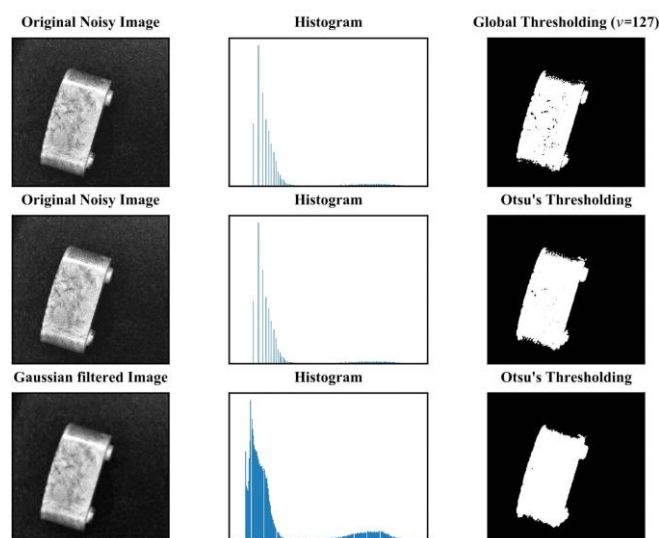
\*Corresponding author. Hongjun Wang, PhD, Professor, research interest: agricultural engineering and machine vision, South China Agricultural University, Guangzhou 510642, China. Tel: +86-18665686187, Email: xtwhj@scau.edu.cn.

conditions, and also proved the feasibility and effectiveness of the algorithm.

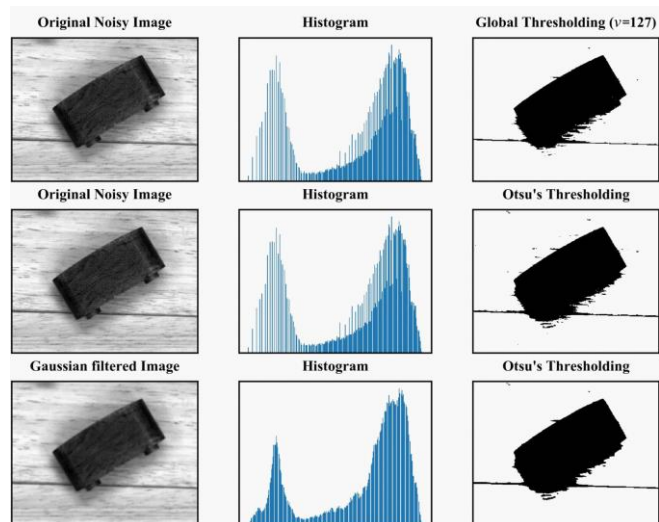
## 2 Materials and methods

### 2.1 Improved Otsu segmentation threshold method

Otsu derives the optimal segmentation threshold based on the gray histogram<sup>[17,18]</sup> and constructs a two-dimensional histogram using the applicable original image and its domain smooth image. When the target image segmentation with bimodal histograms, the gray histogram is divided into two parts with optimal threshold and variance between two parts will be maximized<sup>[19]</sup>. Method of maximal inter-class variance selection used to gradually take optimal threshold  $k$  from  $[0, 255]$  and select  $k$ -value that maximizes the variance between classes as the threshold. Figure 1 shows two examples of threshold segmentation of Otsu in a noisy background which with a global threshold ( $v=127$ ) and Gaussian filter comparison.



a. Simple background target



b. Complicated background target

Figure 1 Otsu performance with global threshold ( $v=127$ ) and Gaussian filter comparison

Otsu algorithm generalized from a single threshold to a multi-threshold, but the essence of the algorithm is still an exhaustive method with very high time complexity. When the traditional multi-threshold Otsu method calculates the optimal threshold, it takes a long time to find the optimal threshold for the  $L^N$  sub-exhaustive traversal calculation of the gray space<sup>[20,21]</sup>.

Therefore, if the threshold search range can be reduced, the segmentation efficiency can be improved. Suppose that the optimal threshold  $T$  ( $T_1, T_2, T_3, \dots, T_{n-1}$ ) has a mathematical correspondence with the mean value of the divided  $N$  types:  $\mu_0,$

$$\mu_1, \dots, \mu_{n-1} \text{ with } T_1 = \left(\frac{1}{2}\right) * (\mu_0 + \mu_1), T_2 = \left(\frac{1}{2}\right) * (\mu_1 + \mu_2) \dots;$$

$$T_{n-1} = \left(\frac{1}{2}\right) * (\mu_{n-2} + \mu_{n-1}).$$

In the process of finding this optimal threshold, it is then exhaustive to calculate the inter-class variance corresponding to all thresholds but to find a set of thresholds that satisfy the multi-threshold Otsu criterion as the optimal threshold. However, the multi-threshold criterion requires the number of iterations  $L^N$ , and the efficiency of the algorithm iteration improves as  $L^N / (T_1, T_2, T_3, \dots, T_n)$ . As the number of thresholds increases,  $T_1, T_2, T_3, \dots, T_n$  will become closer and lower to grayscale, and greater efficiency improvement<sup>[22]</sup>. Let the gray levels of target images have  $L$  levels, and the number of pixels in  $i$ -th level is  $n_i$ , the distribution interval in level  $i$  is  $[0, L-1]$ , and the total number of pixels is counted as:

$$N = \sum_{i=0}^{L-1} n_i \tag{1}$$

Among them, the statistical probability of  $i$ -th level is:

$$P_i = \frac{n_i}{N} \tag{2}$$

Threshold  $k$  was set to divide the pixel into two categories: Target A and background B. The gray level distribution of targets is  $[0, k-1]$ , and the gray level distribution interval of the background is  $[k, L-1]$ . The average gray level of images can be calculated as:

$$\mu = \sum_{i=0}^{L-1} iP_i \tag{3}$$

The proportion of A and B pixels is

$$\omega_0 = \sum_{i=0}^{k-1} P_i, \omega_1 = \sum_{i=k}^{L-1} P_i = 1 - \omega_0 \tag{4}$$

A class and B class pixel average gray is

$$\mu_0 = \frac{\mu_0(k)}{\omega_0}, \mu_1 = \frac{\mu_1(k)}{\omega_1} \tag{5}$$

Among them,

$$\mu_0 = \sum_{i=k}^{L-1} iP_i, \mu_1(k) = \sum_{i=k}^{L-1} iP_i = 1 - \mu_0(k) \tag{6}$$

For an image with a gray level of  $L$ , when calculating the average gray value of a domain matrix of  $N*N$  at each pixel, a definition function of the domain smooth image with respect to  $x, y$  is defined. Assuming function  $L(x, y)$ , let  $F_{ij}$  represent the number of pixels in the image whose gray value is  $i$ , and the pixel whose field average gray value is  $j$  appears in the same spatial position.  $K$  is the square field width of the pixel, and the gray level function  $F_{ij}(m, n)$  can be expressed as:

$$F_{ij} = \frac{1}{k^2} \sum_{i=(k-1)/2}^{(k-1)/2} L(m+i, n+j) \tag{7}$$

It is assumed that the grayscale segmentation threshold is minimum variance  $p$  obtained from the average threshold of the pixel points, and the domain grayscale means segmentation threshold is the value of the two-dimensional joint probability distribution  $q$ <sup>[23,24]</sup>. Then the algorithm uses the ratio to express the background and the detected target.

$$\omega_0 = \sum_{i=1}^p \sum_{j=1}^q F_{ij} = \omega_0(p, q) \tag{8}$$

$$\omega_1 = \sum_{i=q+1}^L \sum_{j=p+1}^L F_{ij} = \omega_1(p, q) \tag{9}$$

In summary, can get the equation as follows:

$$\mu_l(p, q) = (\mu_{i+1}, \mu_{j+1})^T = \left[ \frac{\sum_{i=j}^p \sum_{j=1}^q i F_{ij}}{\omega_0(p, q)}, \frac{\sum_{i=j+q}^p \sum_{j=1+p}^q j F_{ij}}{\omega_0(p, q)} \right] \quad (10)$$

Its derivation formula for variance between classes is:

$$\sigma^2(k) = \omega_0(\mu_0 - \mu)^2 + \omega_1(\mu_1 - \mu_2)^2 = \omega_1\omega_2(\mu_0 - \mu_1)^2 \quad (11)$$

It is assumed that there is only one threshold  $T_x$  each time image segmentation is performed so that the intra-class variance is minimized. For any  $T_x$ , here have  $T \neq T_x$  and  $\sigma^2(T) < \sigma^2(T_x)$ . According to Equation (11), after the first threshold segmentation, the images are classified into two categories based on the class variance and the gray mean. The first split threshold is  $T_x$ , and the two gray scales are  $\mu_1$  and  $\mu_2$  respectively. Making

$T_x = \frac{q}{p} \omega_1 \omega_2 (\mu_1 + \mu_2)$ , for any  $T_N$ , when  $N \in L(m+1, \dots, m+n, m+n+j)$ , can be derived:

$$2N \leq 2T_x \leq \frac{\mu_1 + \mu_2}{\mu_n} \quad (12)$$

When two or more peak sets are close in distance, the peak set can be expressed as:

$$P_0 = \left\{ P_i, P_i \geq P_{(i-1)} \& P_i \geq P_{(i+1)} \& h(l) > \left(\frac{1}{L} * \theta\right) \right\} \quad (13)$$

where,  $h(l)$  original value of the histogram and  $L$  is the number of gray levels, which  $\theta$  is the scale factor. Assuming that the gray levels of the two peaks are  $i$  and  $j$ , when its valleys satisfy a certain value, it can be considered that the thresholds of the two valleys in the histogram can be combined, and the calculation between the two valleys can be expressed as:

$$\text{vally} = \frac{\frac{h(i) + h(j)}{j - i + 1}}{\frac{h(i) + h(j)}{2}} \quad (14)$$

According to Equation (10), in general  $N$  and  $\frac{p}{q}$  are unknown, the pixel gray mean is only the maximum likelihood estimate of the class variance prediction set. Can be drawn:

$$\sigma_x^2(T) = \frac{\sum_{i=j}^p \sum_{j=1}^q i F_{ij}}{\omega_0(p, q)} + \frac{\sum_{i=j+q}^p \sum_{j=1+p}^q j F_{ij}}{\omega_0(p, q)} + \omega_0(\mu_0 - \mu)^2 + \omega_1(\mu_1 - \mu_2)^2 \quad (15)$$

$$\sigma^2(T) < \sigma^2(T_x) \quad (16)$$

It can be proved from the above formula that for any threshold of the grayscale division value  $T$ , the intra-class variance is greater than or equal to the two types of average values. When  $T \neq T_x$  and  $\sigma^2(T) < \sigma^2(T_x)$ , multiple thresholds are minimized within the class. The variance is a measure of the degree of data dispersion if the edge data of the two types of variance is closer to the Otsu threshold, and this threshold is closer to the center distance of the two classes, then two types of the variance of the Otsu threshold segmentation are also close. The Otsu improvement method proposed in this paper will make the threshold point as close as possible to side with large variance, and divide the pixels of the detection target into classes with the smaller variance within the class according to maximum likelihood estimation. If the gray level of the detection target has been close to the pixel gray value of the background, the improved Otsu method performs a threshold search within a limited range until the maximum class spacing variance and the segmentation threshold have been found.

## 2.2 AdaBoost classifier based on Haar-feature

AdaBoost cascade classifiers have achieved good application value in face recognition, vehicle detection and pedestrian detection. Its core idea of the algorithm is to attach importance to the sample with large prediction error and the weak learner with good performance, and to improve the weight of the sample with poor training effect and the weak learner weight with weak learning ability<sup>[25,26]</sup>. In the initial state, the weight of each sample is the same, and a basic learner  $Q_1(x)$  is trained under this sample distribution. The weight of learner  $Q_1(x)$  and the weight of each sample in the training set are determined according to the predict error of the trained learner on the training set, and its corresponding sample weight should be added to the sample with a large  $Q_1(x)$  prediction error. For samples with small prediction errors, here weight is reduced. Under new sample distribution, the weights  $W_j$  corresponding to the basic learners  $Q_2(x)$  and  $Q_2(x)$  are obtained again. After  $n$  cycles,  $n$  basic learners and corresponding weight vectors  $W_n$  are obtained. Finally, the  $n$  basic learners are combined according to the weight vector  $W$  to obtain a strong learner<sup>[27,28]</sup>. Figure 2 shows the framework of the AdaBoost algorithm.

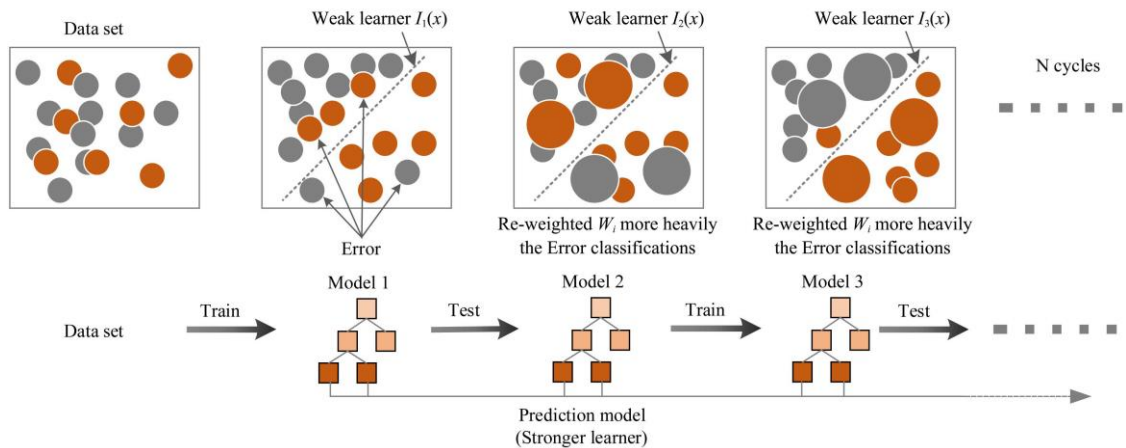


Figure 2 Framework to construct weaker and stronger learners

Haar-feature combined with AdaBoost is often used to extract the structural features when making face detection of humans. It was known that each element of the image contains all pixels and this allows Haar-like feature to compute the sum of rectangular

areas in the image<sup>[29,30]</sup>. Haar features are divided into three categories, edge features, linear features, and specific directional features, combined into feature templates, as shown in Figures 3a-3c.

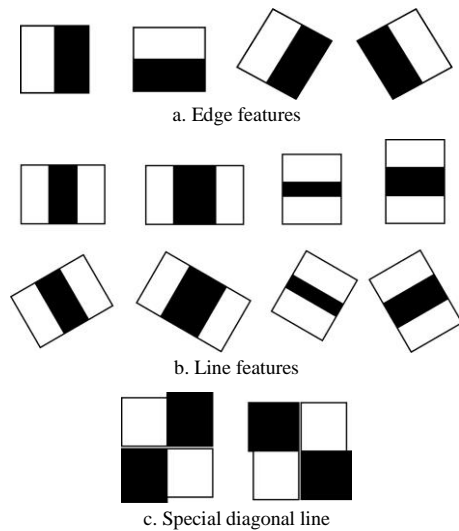


Figure 3 Classification of Haar-features

AdaBoost algorithm forms a stronger classifier by weighting a group of weak classifiers. The purpose is to train a batch of weaker classifiers based on samples and then perform the weighted combination to form a strong classifier<sup>[31,32]</sup>. The construction process of the weak classifier is as follows: Construct a training sample set  $\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$ , where  $x=\{l_1, l_2, \dots, l_k\}$  and  $k$ -dimensional Haar-feature vector extracted from samples,  $y$  represents targets or non- targets and takes a value of 1 or  $-1$ . Set the number of samples belonging to target to  $m$ , and the number of samples not belonging to target as  $n$ . A weaker classifier is trained for each dimension in feature vector extracted from samples, and the corresponding formula is as follows:

$$f_i(x) = \begin{cases} 1 & P_i l_i(x) > P_i \theta_i \\ -1 & \text{other} \end{cases} \quad (17)$$

where,  $l_i(x)$  representing the  $i$  level dimension of Haar eigenvector. Refer to error formula defined by AdaBoost algorithm<sup>[33,34]</sup>:

$$\begin{aligned} \varepsilon_i &= \frac{1}{4} \sum_{j=1}^n \omega_j (f_j - y_j)^2 = \frac{1}{4} \sum_{j=1}^l \omega_j (\gamma^2 + y_i^2) + \\ &\frac{1}{2} \gamma (\sum_{j=1}^n \omega_j y_j - \sum_{j=1}^l \omega_j y_j) \end{aligned} \quad (18)$$

where,  $J$  is subscript of the element in a feature vector;  $N$  is the number of samples, and  $W_j$  is the  $j$ th sample weight. Then there is

$$\Omega = \min(\varepsilon_1, \varepsilon_{-1}) \quad (19)$$

where,  $l_i(x)$  representing the  $i$  level dimension of Haar eigenvector, the value  $P_i$  of the formula is 1 or  $-1$ , and the definition is as follows:

$$\theta_i = \frac{ma_1 + ma_2}{m + n} \quad (20)$$

The  $a_1$  indicates the average of  $i$ -dimensional features of the target, and the  $a_2$  represents the average of  $i$ -dimensional features of the non-target samples. The judgment of  $p_i$  is as follows:

$$P_i = \begin{cases} 1, & a_1 > \theta_i \\ -1, & \text{other} \end{cases} \quad (21)$$

Therefore, the construction process of the weak classifier is expressed as, when  $a_1$  and  $l_i(x)$  is less than or greater than  $\theta_i$  simultaneously. The predicted sample can be judged as the target, otherwise, it is determined to be the non-target<sup>[35,36]</sup>. The training process for the strong classifier is as follows:

(1) Prepare a training set  $D=\{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$  where  $x=\{l_1, l_2, \dots, l_k\}$  represents the  $k$ -dimensional feature vector extracted from the training sample, and  $y$  is the category label of the sample, and its value set as 1 or  $-1$ . Corresponding to target

and non-article, the number of target samples is  $n_1$ , and the number of non- targets is  $n_2$ .

(2) The weight of each sample in the training set is initialized as:

$$\omega_{i,j} = \begin{cases} \frac{1}{2n_1} & y = 1 \\ \frac{1}{2n_2} & y = -1 \end{cases} \quad (22)$$

(3) Setting  $T$  to the maximum number of training rounds, in each round of training, the sample weights are normalized first. For each feature  $L$ , a weak classifier is constructed as  $f_L(x)$  and its error rate  $\varepsilon_i$  is calculated. From which the weak classifier with the smallest error rate is selected as the optimal classifier  $f_i(x)$  of the round, and the error rate  $\varepsilon_i$  is calculated. And the weight of the optimal classifier is derived according to the error rate:

$$\alpha_i = \frac{1}{2} \ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right) \quad (23)$$

According to the classification of the optimal classifier, the training set sample update distribution is:

$$\omega_{i+1,j} = \omega_{i,j} \beta_i^{1-e^j} \quad (24)$$

$$\beta_i = \frac{\varepsilon_i}{1-\varepsilon_i} \quad (25)$$

Among them, when the training samples are classified correctly, the equation  $e^j=1$  holds; when classified incorrectly, the equation  $e^j=0$  holds.

(4) Repeat each round of training and continuously update the weights until the number of training rounds reaches  $T$ . The final strong classifier formula is as follows:

$$H(x) = \text{sign}\left(\sum_{i=1}^T \alpha_i f_i(x)\right) \quad (26)$$

### 2.3 Construction of training sets and extract contour features

In this experiment, to collect 600 background images containing interfering objects into a training set, part of them showing in Figure 4 and Figure 5. Here contained target images with various poses for weaker classifier training. True positive rate ( $t_p$ ) and false positive rate ( $f_p$ ) were used as predicted indicators to evaluate the target recognition rate. The first step is to calculate the anti-interference error rate of targets in different contexts, and the second step is the recognition integrity rate of the target in the case that it can be identified<sup>[37,38]</sup>. Figure 5 shows the samples of targets at different angles. It should be noted that our goal is to input 3D-pose images of the targets and output its mathematical parameters of 3D-pose, such as the plane normal vector and Euler angle in the world coordinate system.



Figure 4 Complex background training

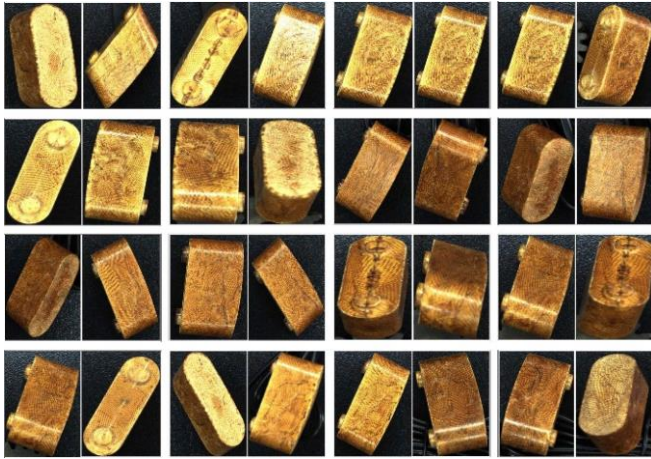


Figure 5 Strong classifier classification results

In summary, we implement image binarization processing with Otsu segmentation and tracking target contours. It was able to identify the shape, the texture of the target<sup>[39,40]</sup>. At this stage, we introduce the Hough transform for pyramid search and region of interest extraction. Then output a rough outline of the target. Finally, the dimension reduction of the target is input to the AdaBoost classifier for training (Figure 6).

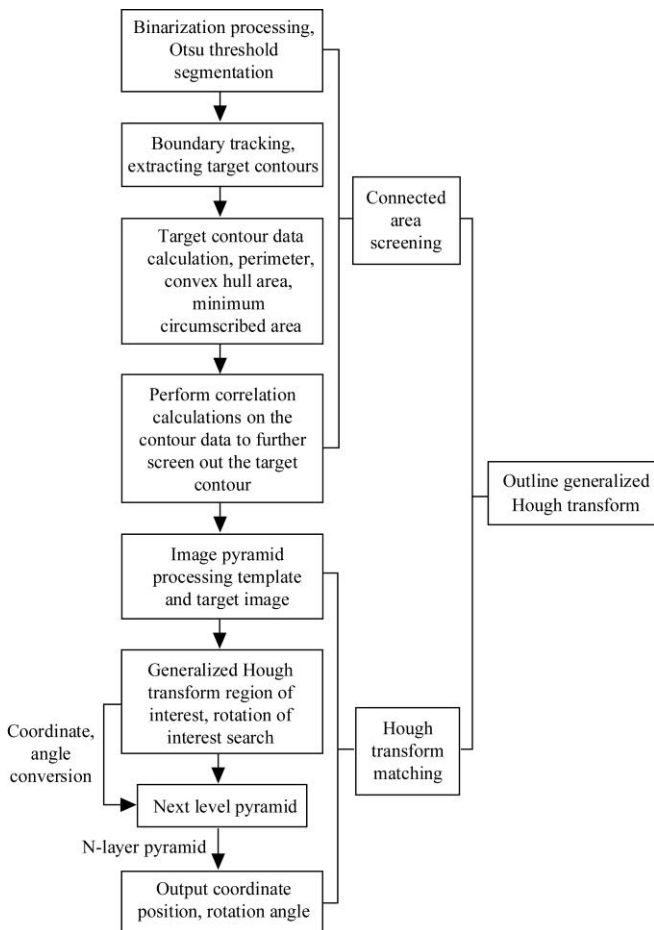


Figure 6 Improved Otsu with generalized Hough transform algorithm framework

### 3 Results and discussion

#### 3.1 Otsu threshold segmentation effect and AdaBoost classification accuracy analysis

The target image acquisition device is composed of four parts, acquisition hardware, bracket, light source, laser range finder and

image processing system<sup>[41,42]</sup>. The image acquisition system was assembled by a CCD camera-Vie MV-EM200C (resolution 1600×1200, frame rate 40 fps). The illumination source consists of four OPT-LI9022 24V-2.7W LED lamps. Laser range finder use BOSCH-GML25. Experimental targets are wooden carcass, which its length is 49.32 mm, width is 17.00 mm and height is 23.40 mm; the side small cylinder diameter is 7.80 mm. The image processing system uses VS2013+OpenCV3.0 and MATLAB graphics modules. Measuring the lens distortion parameters at different positions before each experiment, it can be seen that most of the parameters can negligible according to the accuracy requirements of the experiment<sup>[43,44]</sup>. Suppose the geometric coordinates of point  $(x, y)$  on the image plane under the camera coordinate system are  $\varepsilon_x(x, y)$  and  $\varepsilon_y(x, y)$ . Due to the camera calibration accuracy considering nonlinear distortion, here need to avoid too many nonlinear parameters to cause the instability of the equation solution. So when calculating the polynomial of the distortion model, such as  $\varepsilon_x(x, y) = x*(k_1r^2 + k_2r^4)$  and  $\varepsilon_y(x, y) = y*(k_1r^2 + k_2r^4)$ , radial distortion generally takes first or two parameters, so only two parameters  $k_1$  and  $k_2$  need to be considered in the calculation. Here  $k_1$  and  $k_2$  are the radial distortion parameter that needs to be calibrated, and  $r$  is the distance of the point from the imaging center. The target image converted from the RGB model to the  $YCbCr$  model. When there is no overlap and the discrimination is obvious, the OSTU segmentation algorithm can be used to calculated better detection results. After the original color picture is converted into the  $YCbCr$  mode, the background and the target have a large difference in the  $C_b$  mode. The color of the surface of different targets is different<sup>[45,46]</sup>. It is difficult to meet the requirements of actual detection by using a fixed threshold. The visualization of the three channels is shown in Figure 7. The threshold split histogram is shown in Figure 8.

There can easily get a conclusion from Figure 8b that its histogram has distinct peaks between background and detected targets and a distinction between the thresholds and valleys. Although Figure 8a and Figure 8c also have two peaks, the histogram threshold segmentation is not obvious enough. The distance between the two peaks is too close will make the image noise affect the segmentation effect.

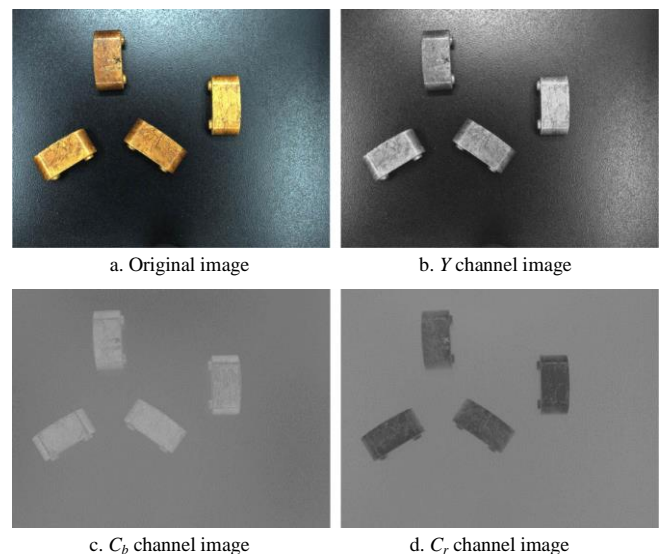


Figure 7 Three-channel visualization of the  $YCbCr$  of the target image

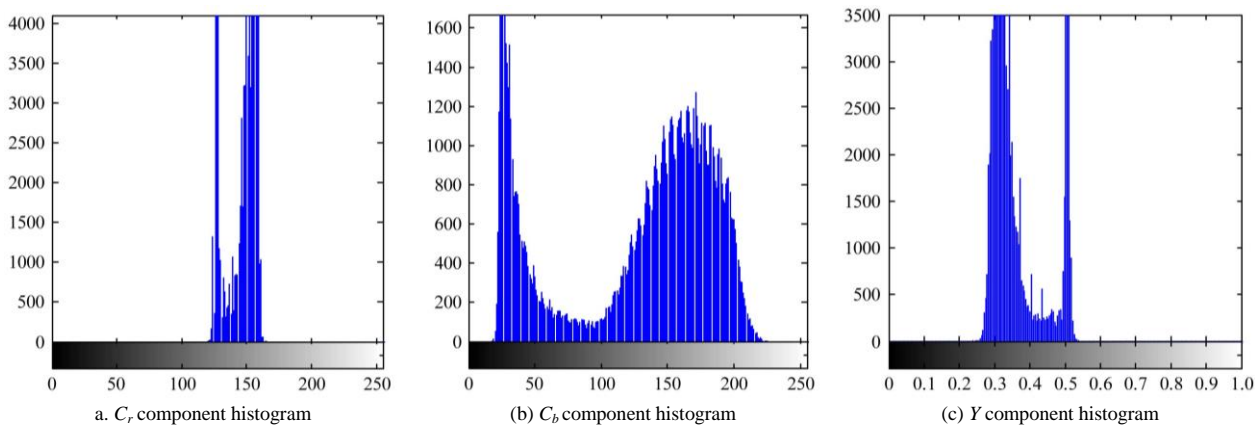
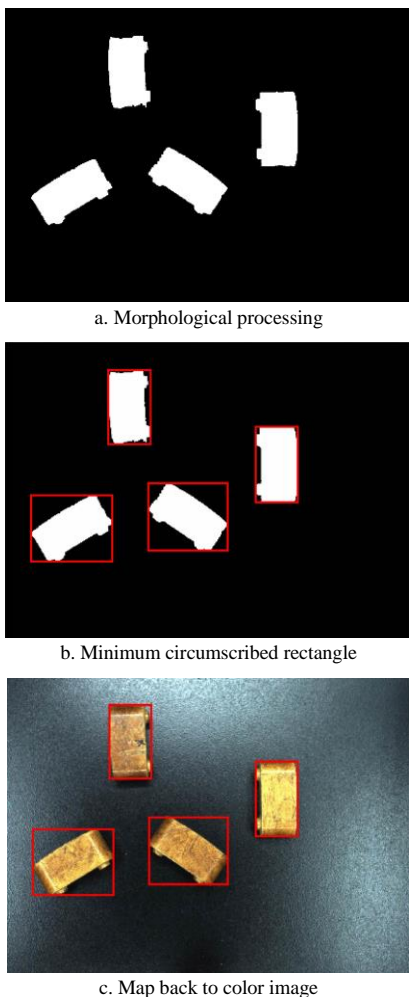


Figure 8 Threshold split histogram

Therefore, the Otsu algorithm with or without Hough transform was used to automatically extract the threshold and segment the  $C_b$  channel image. The experimental results show that the algorithm has a good segmentation and localization effect for non-overlapping scenes. In the absence of a non-target background, the specific process of threshold segmentation based on the  $C_b$  channel is shown in Figure 9.



c. Map back to color image

Note: Original image of the target is acquired, then converted to grayscale image. The  $C_b$  channel is extracted by the Otsu algorithm, and appropriate segmentation threshold is set according to the histogram. Then calculate the minimum boundary pitch and return the image to RGB mode

Figure 9 Threshold segmentation process

After the AdaBoost classifier had been trained, scanned detection was performed on the test image. The multi-scale strategy is used to detect the image so that the positioning results

are more accurate, and the region merging technique is needed to solve the detection window coincidence problem. The image acquisition platform is used to capture 600 sample images of the target, covering different rotation angles as much as possible<sup>[49,50,51]</sup>. With the 2000 sheets of non-targets samples, those samples are as diverse as possible. The data sets contain scenes such as animals, trees, landscapes and roads. Configuring Microsoft Visual Studio 2013+Opencv3.0 experimental platform under Windows, then training AdaBoost cascade detector based on Haar feature<sup>[47,48]</sup>. The training layer is set to 18, the minimum hit rate is set to 0.95, and the maximum false detection rate is set to 0.5. Multiple targets are randomly placed, but they do not intersect with each other.

**3.2 Complex background without overlap testing**

It can be seen from Table.1 that there are three different algorithms which include hough contour recognition, cascade AdaBoost and Haar-feature training based on identifies results of targets. In this step, our testing operating system of the computer is windows 10 and its processor model is Intel(R) Core(TM) i5-8250U@1.60GHz. The recognition efficiency of Hough contour transform in the complex background is low, only 87.44% and its false recognition rate reach 10.94%. In the CB and FB backgrounds, the training times for the cascade-based Adaboost method are 52.17 h and 32.71 h, respectively. Based on the Haar feature training method, better model test results were obtained. Its training time was 55.83 h and 28.44 h, respectively, and the recognition time was relatively shortened to 52 ms and 31 ms. The correct sample identification rate reached 97.13% and 98.42% and the misrecognition rate corresponds to 4.48% and 3.91%. Experiments show that AdaBoost based on Haar-feature training is an ideal target classification mathematical model and can meet the target positioning test proposed in this study.

**Table 1 Model test results**

Algorithm	Sample	Training $T_s/h$	Recognition $T_i$	$t_p$	$f_p$
Hough Contour Recognition	CB	/	45 ms	87.44%	10.94%
	FB	/	1 min 32 ms	90.75%	8.87%
Cascade AdaBoost	CB	52.17	1 min 02 ms	94.18%	7.11%
	FB	32.71	39 ms	93.53%	5.34%
Haar-feature training	CB	55.83	52 ms	97.13%	4.48%
	FB	28.44	31 ms	98.42%	3.91%

Note: CB: complex background; FB: the background that does not contain non-target.

**3.3 Overlap and Non-overlap testing**

In order to verify the effectiveness of the proposed method,

AdaBoost training samples are divided into overlapping backgrounds and no overlapping background for weak trainer debugging. The algorithm can separate overlapping artifacts one by one. However, the color difference between the targets is small and overlaps each other. Compared with the Otsu segmentation algorithm, the AdaBoost algorithm is better for the segmentation test of complex scenes. Simulate the target gripping process and manually grip the one that is placed the easiest to get normal vectors. The recognition effect before or after the gripping action is shown in Table 2. It can be seen from the experimental data that the different threshold division ratios do not affect the recognition target time of the algorithm. A non-overlapping target training set has a significant accuracy

advantage. The training time is fastest at 30.50 ms, and the accuracy rate is 98%. In the classification recognition scene with overlapping targets, according to rectangular area statistics when the target is recognized, its accuracy of the recognition by the training algorithm is lower for the target with the larger inclination angle. For example, the quadrilateral area in Table 2 is only 309.88 mm<sup>2</sup>, which only achieves an accuracy of 79%. In the identifying process and clamping the target one by one process, manually clamping the flattest part which changed from a complex scene to a simple scene. The overlap of the target will be gradually reduced, and the original occlusion is not identified targets will be exposed, making the segmentation performance better, and the recognition rate will be higher.

**Table 2 Overlap and non-overlapping background test results**

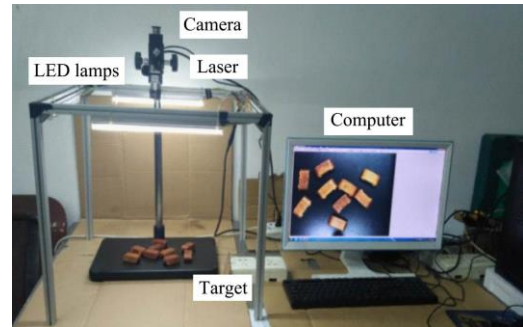
Algorithm	Fraction threshold	Grayscale threshold	Recognition $T_i$	$L_s/mm$	$W_s/mm$	$L_s*W_s$	Accuracy
Overlap scene	0.45	125	50.34 ms	38.17	17.22	657.28	87%
	0.32	167	45.11 ms	27.55	12.31	339.14	92%
	0.47	181	45.05 ms	42.15	22.08	996.91	96%
	0.26	132	1 min 32 ms	16.54	8.85	146.37	89%
	0.44	137	1 min53 ms	21.64	14.32	309.88	79%
	0.38	166	57.31 ms	28.96	17.41	504.19	90%
Non-overlap scene	0.27	152	34.4 1ms	45.21	20.17	911.88	91%
	0.53	137	35.09 ms	37.64	21.66	815.28	95%
	0.49	180	47.28 ms	13.20	4.98	65.73	87%
	0.31	140	30.50 ms	27.41	7.92	217.08	98%
	0.22	188	1 min 11 ms	10.33	6.09	62.90	90%
	0.48	193	48.54 ms	18.59	13.22	245.75	95%

### 3.4 Classification verification experiment based on normal vector and monocular system

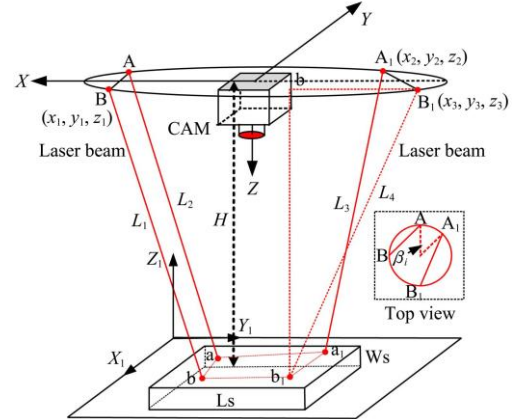
The target identified by AdaBoost needs to be solved using mathematical geometry modeling and transfer the target's three-dimensional orientation parameters to the computer. The camera coordinate system is shown in Figure 10a. The red line in the figure below represents the laser rangefinder light, and those points had been marked. After the coordinate system established, let point A and point B be the original position of two laser range finder,  $A_1B_1$  is the position after the middle rail rotates by a certain angle, and the corresponding points of the two laser range finder hit the target are a, b,  $a_1$ ,  $b_1$ . It is assumed that the angle of rotation of the middle rail of A to  $A_1$  is  $\beta$  (Top view in Figure 10b), and the deflection angles of the laser range finder corresponding to the four points A,  $A_1$ , B,  $B_1$  are  $\alpha$ ,  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$ .

Establishing the coordinate system and making assumptions about each parameter, we can obtain the coordinates corresponding to each laser spot in the coordinate system. Take the two points  $A_1$  and  $A_2$  as an example. Firstly, solving the coordinates of point A,  $L$  is the measured result of the laser range finder,  $\alpha$  is the deflection angle of the alpha laser range finder,  $H$  is the vertical distance of laser point to the coordinate system, or can be understood as the  $z$  coordinate corresponding to point A. Point M is the  $X$  coordinate of point A. Since point A is the original position,  $y$  coordinate of point A was zero. Showing by the graph, the left side of point A( $x, y, z$ ),  $Z$  equal to  $H$ , and  $H=L*\sin\alpha$ ,  $X=R-L*\cos\alpha$ ,  $Y=0$  can be found. Solving the  $A_1$  coordinate,  $L_1$  is the measured result of the laser range finder.  $\alpha_1$  is the deflection angle of the laser range finder, and  $H_1$  is the vertical distance from the laser point to coordinate system, that is, the  $Z_1$  coordinate corresponding to the  $A_1$  point. Due to the rotation angle of point A to the middle rail of  $A_1$  was  $\beta$ , and let  $A_1$  take the other

coordinates  $x_1, y_1$ . As shown in the figure, solving the  $A_1$  point coordinates ( $x_1, y_1, z_1$ ) according to the trigonometric function relationship. The coordinates of  $Z_1$  can be calculated according to equation  $Z_1=L_1*\sin\alpha_1$ . The  $X_1$  and  $Y_1$  can be calculated according to their trigonometric relationship.



a. Experiment platform



b. Experimental monocular system design

Figure 10 Monocular vision and laser experiment system

In summary, by analyzing the laser spot, the coordinates of each laser spot on the coordinate system can be obtained. It is assumed that the coordinates corresponding to the four points A, A<sub>1</sub>, B, and B<sub>1</sub> are as follows: A(x, y, z), A<sub>1</sub>(x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>), B(x<sub>2</sub>, y<sub>2</sub>, z<sub>2</sub>), B<sub>1</sub>(x<sub>3</sub>, y<sub>3</sub>, z<sub>3</sub>). Assuming that the spatial coordinates of the four laser points A, A<sub>1</sub>, B, B<sub>1</sub> in the coordinate system are A(x, y, z), A<sub>1</sub>(x<sub>1</sub>, y<sub>1</sub>, z<sub>1</sub>), B(x<sub>2</sub>, y<sub>2</sub>, z<sub>2</sub>), B<sub>1</sub>(x<sub>3</sub>, y<sub>3</sub>, z<sub>3</sub>). The four laser spots are on the target plane, so two vectors on this plane can be obtained. Finding the angle between two planes can be indirectly obtained by finding the angle between two plane normal vectors, and the plane normal vector is innumerable. Therefore, the normal vector *n* of the surface can be set as (x<sub>0</sub>, y<sub>0</sub>, 1), and have the quantitative relationship of *n*·AA<sub>1</sub>=0, *n*·AB=0. Solving the equation, it can be found the specific coordinates of x<sub>0</sub> and y<sub>0</sub> as shown in the following Equation (27).

$$\begin{cases} x_0(x_1 - x) + y_0(y_1 - y) + (z_1 - z) = 0 \\ x_0(x_2 - x) + y_0(y_2 - y) + (z_2 - z) = 0 \\ x_0 = \frac{y_0(y_1 - y) + z_1 - z}{x - x_1} \\ y_0 = \frac{(z - z_1)(x_2 - x) + (z - z_2)(x - x_1)}{(x_2 - x)(y_1 - y) + (y_2 - y)(x - x_1)} \end{cases} \quad (27)$$

From this, the normal vector *n*(x<sub>0</sub>, y<sub>0</sub>, 1) of the target surface can be worked out. Therefore, the normal vector of the horizontal plane can be set as *a* (0, 0, 1). If the angle between the two planes is *θ*, then

$$\sin \theta = \frac{\vec{n} \cdot \vec{a}}{|\vec{n}| |\vec{a}|}, \quad \theta = \arcsin \frac{\vec{n} \cdot \vec{a}}{|\vec{n}| |\vec{a}|} \quad (28)$$

The inverse trigonometric function (28) can be used, and the angle between the target and horizontal plane is *θ*. Figure 11 shows the distribution statistics of the four sets of experiments at different angles of the target. During the experiment, four points are used as a set of data, and ten sets of data are respectively measured, BC represents before correction, and AC represents after correction. The data is as in Table 3. Combining with Table 3 and Figure 11, it is obvious that in the training set, the target of this paper is more likely to produce errors when the tilt angle is larger, and the statistical result is less close to the prediction set. However, AdaBoost feature training can get correction errors close to 1.076 and 1.045. This shows that Haar-based feature training has a very high accuracy rate with actual measurement results.

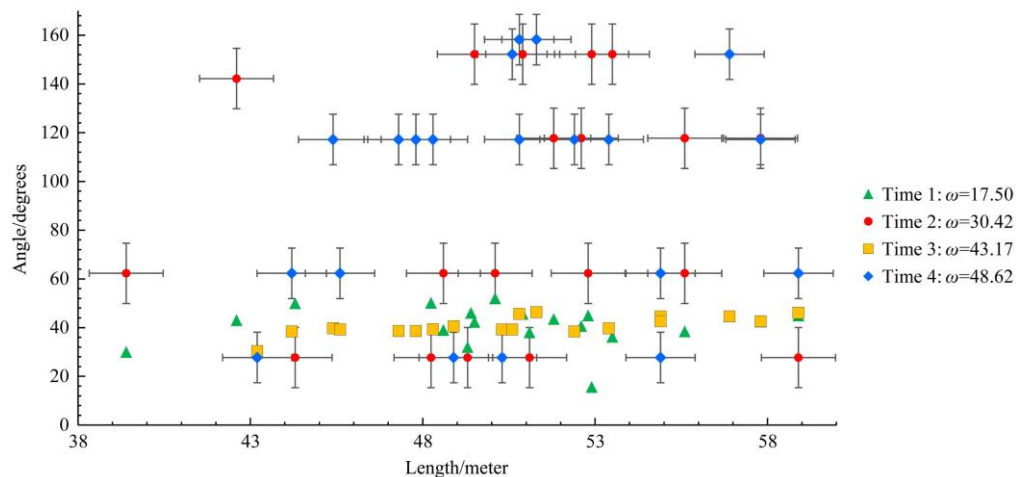


Figure 11 Statistics of target 3D attitude detection experiment results with error lines

Table 3 Four kinds of proposed methods for angle prediction

Test No.	Otsu Algorithm				Improved algorithm Otsu				Hough transform				AdaBoost Classifier			
	<i>θ</i> (BC)	<i>θ</i> (AC)	E <sub>BC</sub>	E <sub>AC</sub>	<i>θ</i> (BC)	<i>θ</i> (AC)	E <sub>BC</sub>	E <sub>AC</sub>	<i>θ</i> (BC)	<i>θ</i> (AC)	E <sub>BC</sub>	E <sub>AC</sub>	<i>θ</i> (BC)	<i>θ</i> (AC)	E <sub>BC</sub>	E <sub>AC</sub>
1	15.183	16.259	2.640	1.507	18.532	19.245	2.064	1.507	22.986	24.152	2.640	1.491	28.307	28.622	1.726	1.384
2	7.939	8.624	2.071	1.380	22.453	21.138	2.456	1.138	12.911	13.952	22.099	1.103	33.286	33.954	1.723	1.045
3	43.121	43.502	1.885	1.487	38.100	38.827	1.904	1.183	16.778	16.745	1.423	1.491	40.786	40.572	1.523	1.076

It can be seen from the above table that the algorithm error range was within 1.48 to 1.88 units, the angle between the target and horizontal plane was *θ* and its error not exceeded 0.8 units. Reading angle accuracy was within 0.05°. Through measurement data after Hough transform, the normal vector of the regular target in the camera coordinate system and its positional relationship in space can be calculated. The existence of error may be related to the positional relationship between the laser range finder and camera. The Center axis of the camera does not coincide with the center of the annular guide. Since the camera may have some defects during the production process, the center point of the camera lens may not be on the optical axis with the center point as a whole. Even if we performed camera calibration during the experiment, the parameters of the camera are recognized, but during the installation process. It is still difficult to ensure that the

center axis coincides with the center point of the ring guide, resulting in a certain error. 10 sets of different data were used to measure the accuracy of SPSS detection. As shown in Figure 12 below, the black line is an accuracy fitting curve of improved detectors that Otsu combined with Haar-feature in AdaBoost. Under normal enhanced training, the black one of the accuracy of angular standard deviation is obviously better than the red and blue dotted line. The Otsu algorithm can accurately segment the thresholds and perform variance-like statistics, but in many modern classification methods, relative loss problems occur even if optimized to better classifiers. When the algorithm performs learning and efficient feature selection, AdaBoost can make up for this shortcoming. At the same time as the weaker classifier was preferred, the high detection rate of the stronger classifier is improved, and the goal of each classifier in the cascade is not a low



error. Therefore, the AdaBoost classifier of Otsu class variance segmentation and Haar feature training can be used in monocular visual 3D pose recognition.

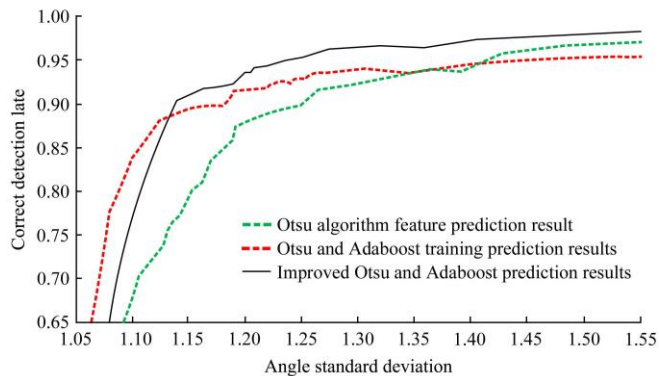


Figure 12 Algorithm accuracy prediction curve

## 4 Conclusions

(1) The Otsu class variance threshold derives the optimal segmentation value based on the gray histogram of regular targets, and divided the gray histogram into two parts with the optimal threshold or maximizes the variance between detection targets.

(2) Using Haar-feature extraction in training can improve the accuracy of AdaBoost, the  $N$  basic learners and the corresponding weight vector  $W_n$  are inferred through  $n$  cycles. The weight vector  $W$  combined  $n$  basic learners to obtain a strong learner and achieved a classification effect of 98%.

(3) Comparing a number of different methods, the designed monocular system test results showed that the 3D pose detection of the rule target could use the machine learning training algorithm to obtain the superior effect.

(4) Experiments have shown that even if there was distortion in the lens, the calculation error of no more than 1.48 was obtained. This study systematically combined machine learning algorithms and 3D geometric mathematical models and proposed a complete monocular visual 3D orientation recognition algorithm.

## Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (Grant No. 51705365). The authors also acknowledge the State Key Research Program of China (Grant No. 2017YFD0700404), the Guangdong Provincial Department of Education Project (Grant No. 2016KZDXM027), and the Guangdong Provincial Department of Agriculture (Grant No. 2019KJ129).

## [References]

- [1] Baró X, Escalera S, Vitrià J, Pujol O, Radeva P. Traffic sign recognition using evolutionary adaboost detection and forest-ECOC classification. *IEEE Transactions on Intelligent Transportation Systems*, 2009; 10(1): 113–126.
- [2] Sappa D, Dornaika F, Ponsa D, Gerónimo D, López A. An efficient approach to onboard stereo vision system pose estimation. *IEEE Transactions on Intelligent Transportation Systems*, 2008; 9(3): 476–490.
- [3] Sun J. Improved 2D maximum between-cluster variance algorithm and its application to cucumber target segmentation. *Transactions of the CSAE*, 2009; 25(10): 176–181. (in Chinese)
- [4] Guo H X, Li Y J, Li Y N, Liu X, Li J L. BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification. *Engineering Applications of Artificial Intelligence*, 2016; 49: 176–193.
- [5] Li Z, Yuan Y, Ke F, He W, Su C Y. Robust vision-based tube model predictive control of multiple mobile robots for leader–follower formation. *IEEE Transactions on Industrial Electronics*, 2019; 67(4): 3096–3106.
- [6] Michels J, Saxena A, Ng A Y. High speed obstacle avoidance using monocular vision and reinforcement learning. In: *Proceedings of the 22nd International Conference on Machine Learning*, 2005; pp: 593–600.
- [7] Sund T, Eilertsen K. An algorithm for fast adaptive image binarization with applications in radiotherapy imaging. *IEEE Transactions on Medical Imaging*, 2003; 22(1): 22–28.
- [8] Fan J L, Zhao F. Two-dimensional Otsu's curve thresholding segmentation method for gray-level images. *Acta Electronica Sinica*, 2007; 35(4): 751. doi: 10.1002/jrs.1570. (in Chinese)
- [9] Gong Q, Ni L, Tang P F, Wang F F. Fast three-dimensional Otsu image segmentation algorithm based on decomposition. *Journal of Computer Applications*, 2012; 32(6): 1526–1528. (in Chinese)
- [10] Meechart K, Auethavekiat S, Sa-ing V. An Automatic Detection for Avian Blood Cell based on Adaptive Thresholding Algorithm. In: *2019 12th Biomedical Engineering International Conference (BMEiCON)*, 2019; 11: 1–4.
- [11] Jin L S, Lei T, Wang R B, Guo L, Chu J W. An improved Otsu image segmentation algorithm for path mark detection under variable illumination. In: *IEEE Proceedings. Intelligent Vehicles Symposium*, 2005; pp.840–844.
- [12] Qiao N, Sun P. Study of improved Otsu algorithm and its ration evaluation analysis for PCB photoelectric image segmentation. *Optik*, 2014; 125(17): 4784–4787.
- [13] Qin J, Shen X, Mei F, Fang Z. An Otsu multi-thresholds segmentation algorithm based on improved ACO. *The Journal of Supercomputing*, 2019; 75(2): 955–967.
- [14] Zhou M, Wei H. Face verification using gaborwavelets and adaboost. In: *18th International Conference on Pattern Recognition (ICPR'06)*, 2006; 1: 404–407.
- [15] Lu H, Wang H, Yoon S W, Won D. Real-Time stencil printing optimization using a hybrid multi-layer online sequential extreme learning and evolutionary search approach. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 2019; 9(12): 2490–2498.
- [16] Tian H X, Mao Z Z. An ensemble ELM based on modified AdaBoost RT algorithm for predicting the temperature of molten steel in ladle furnace. *IEEE Transactions on Automation Science and Engineering*, 2009; 7(1): 73–80.
- [17] Merzbach M H, Elbayoumi M. Efficient solution of Otsu multilevel image thresholding: A comparative study. *Expert Systems with Applications*, 2019; 116: 299–309.
- [18] Bhandari A K, Kumar I V. A context sensitive energy thresholding based 3D Otsu function for image segmentation using human learning optimization. *Applied Soft Computing*, 2019; 82: 105570. doi: 10.1016/j.asoc.2019.105570.
- [19] Khairuzzaman A K, Chaudhury S. Brain MR image multilevel thresholding by using particle swarm optimization, Otsu method and anisotropic diffusion. *International Journal of Applied Metaheuristic Computing (IJAMC)*, 2019; 10(3): 91–106.
- [20] Sundaram R, Jayaraman P, Rangarajan R, Rengasri R, Rajeshwari C, Ravichandran K S. Automated optic papilla segmentation approach using normalized Otsu thresholding. *Journal of Medical Imaging and Health Informatics*, 2019; 9(7): 1346–1353.
- [21] Wang W L, Duan L M, Wang Y. Fast image segmentation using two-dimensional Otsu based on estimation of distribution algorithm. *Journal of Electrical and Computer Engineering*, 2017; 2017(3): 1735176. doi: 10.1155/2017/1735176.
- [22] Campbell J, Sukthankar R, Nourbakhsh I, Pahwa A. A robust visual odometry and precipice detection system using consumer-grade monocular vision. In: *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. Barcelona: IEEE, 2005; pp.3421–3427.
- [23] Caballero F, Merino L, Ferruz J, Ollero A. Unmanned aerial vehicle localization based on monocular vision and online mosaicking. *Journal of Intelligent and Robotic Systems*, 2009; 55(4–5): 323–343.
- [24] Hanif S M, Prevost L. Text detection and localization in complex scene images using constrained adaboost algorithm. In: *2009 10th International Conference on Document Analysis and Recognition*. Barcelona: IEEE, 2009; pp.1–5.
- [25] Kim S, Song W J, Kim S H. Robust ground target detection by SAR and IR sensor fusion using Adaboost-based feature selection. *Sensors*, 2016; 16(7): 1117. doi: 10.3390/s16071117.
- [26] Hu W M, Gao J, Wang Y G, Wu O, Maybank S. Online adaboost-based parameterized methods for dynamic distributed network intrusion detection. *IEEE Transactions on Cybernetics*, 2013; 44(1): 66–82.

- [27] Liu M H. Fingerprint classification based on Adaboost learning from singularity features. *Pattern Recognition*, 2010; 43(3): 1062–1070.
- [28] Gao Y, Liu H, Sun X H, Wang C, Liu Y. Violence detection using oriented violent flows. *Image and Vision Computing*, 2016; 48–49: 37–41.
- [29] Viola P, Jones M. Fast and robust classification using asymmetric Adaboost and a detector cascade. *Advances in Neural Information Processing Systems*, 2002; 14: 1311–1318.
- [30] Liu Y H, Huang S, Huang Y D. Motor imagery EEG classification for patients with amyotrophic lateral sclerosis using fractal dimension and Fisher's criterion-based channel selection. *Sensors*, 2017; 17(7): 1557. doi: 10.3390/s17071557.
- [31] Kaur S, Sharma D A. Multi-classifiers comparison for protein secondary structure prediction. In: 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). Noida: IEEE, 2019; pp.359–364.
- [32] Owusu E, Zhan Y, Mao Q R. A neural-AdaBoost based facial expression recognition system. *Expert Systems with Applications*, 2014; 41(7): 3383–3390.
- [33] Ferri M, Mangili F, Viano G. Projective pose estimation of linear and quadratic primitives in monocular computer vision. *CVGIP: Image understanding*, 1993; 58(1): 66–84.
- [34] Guo L, Ge P S, Zhang M H, Li L H, Zhao Y B. Pedestrian detection for intelligent transportation systems combining AdaBoost algorithm and support vector machine. *Expert Systems with Applications*, 2012; 39(4): 4274–4286.
- [35] Solomatine D P, Shrestha D L. AdaBoost.RT: A boosting algorithm for regression problems. In: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541). Budapest: IEEE, 2004; 2: pp.1163–1168.
- [36] Chaurasia V, Pal S. Skin Diseases prediction: binary classification machine learning & multi model ensemble techniques. *Indian Journal of Public Health Research & Development*, 2020; 11(1): 737–742.
- [37] Bjorkman M, Kragic D. Combination of foveal and peripheral vision for object recognition and pose estimation. In: IEEE International Conference on Robotics and Automation. New Orleans: IEEE, 2004; 5: 5135–5140.
- [38] Loianno G, Kumar V. Cooperative transportation using small quadrotors using monocular vision and inertial sensing. *IEEE Robotics and Automation Letters*, 2017; 3(2): 680–687.
- [39] Yang T, Li P Q, Zhang H M, Li J, Li Z. Monocular vision SLAM-based UAV autonomous landing in emergencies and unknown environments. *Electronics*, 2018; 7(5): 73. doi: 10.3390/electronics7050073.
- [40] Royer E, Lhuillier M, Dhôme M, Lavest J M. Monocular vision for mobile robot localization and autonomous navigation. *International Journal of Computer Vision*, 2007; 74(3): 237–260.
- [41] Yuan W B, Cao Z Q, Zhang Y J, Tan M. A robot pose estimation approach based on object tracking in monitoring scenes. *International Journal of Robotics and Automation*, 2017; 32(3): 256–265.
- [42] Pang M L, Tiberius C. Precision and reliability of tightly coupled PPP GNSS and landmark monocular vision positioning. *Sensors*, 2020; 20(5): 1537. doi: 10.3390/s20051537.
- [43] Solomatine D P, Shrestha D L. 2004, July. AdaBoost.RT: a boosting algorithm for regression problems. In: 2004 IEEE International Joint Conference on Neural Networks, 2004; 2: 1163–1168.
- [44] Ye N, Yang B W, Zhou H C, Zhang L Y. A calibration trilogy of monocular-vision-based aircraft boresight system. *Measurement*, 2018; 117: 133–143.
- [45] Tourmier G, Valenti M, How J, Feron E. Estimation and control of a quadrotor vehicle using monocular vision and moire patterns. In: AIAA Guidance, Navigation, and Control Conference and Exhibit, 2006; 6711. doi: 10.2514/6.2006-6711.
- [46] Carreras X, M àrquez L, Padró L. A simple named entity extractor using AdaBoost. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, 2003; pp.152–155.
- [47] Li X, Liu W, Pan Y, Ma J, Wang F. A knowledge-driven approach for 3D high temporal-spatial measurement of an arbitrary contouring error of CNC machine tools using monocular vision. *Sensors*, 2019; 19(3): 744. doi: 10.3390/s19030744.
- [48] Meng C C, Bao H, Ma Y, Xu X K, Li Y Q. Visual Meterstick: Preceding vehicle ranging using monocular vision based on the fitting method. *Symmetry*, 2019; 11(9): 1081. doi: 10.3390/sym11091081.
- [49] Hu W M, Hu W, Maybank S. Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2008; 38(2): 577–583.
- [50] Hwang S Y, Song J B. Monocular vision-based SLAM in indoor environment using corner, lamp, and door features from upward-looking camera. *IEEE Transactions on Industrial Electronics*, 2011; 58(10): 4804–4812.
- [51] Khammari A, Nashashibi F, Abramson Y, Laugeau C. Vehicle detection combining gradient analysis and AdaBoost classification. In: Proceedings. 2005 IEEE Intelligent Transportation Systems. Vienna: IEEE, 2005; pp.66–71.