Research on steel barrel flattened seam recognition based on machine vision

Hao Niu1, Bowen Wu, Qingyan Wang and Zhisong Zhu

School of Mechanical Engineering, Nantong University, Nantong 226019, Jiangsu, China

1 E-mail: 2063869755@qq.com

Abstract. Aiming at the problems of low efficiency and poor accuracy in manually identifying the flattened weld seam. The method of combining machine vision technology with multi-thread and multi-process is proposed, and the detection and recognition system of flattened weld seam is designed. OTSU adaptive threshold algorithm combined with Canny operator edge detection method is used for image segmentation. The probabilistic Hough transform algorithm is used to extract straight lines to identify flattened weld seam. OpenCV function library and Python are used to implement image processing algorithm, and PyQt5 is used to design human-computer interface to monitor the status of weld seam detection in real time. Meanwhile, a processing pool for parallel processing of recognition algorithm is established, and Modbus communication and image acquisition are realized by multi-thread programming. The experimental results show that the system meets the design requirements, has the characteristics of small amount of calculation and accurate identification, improves the processing efficiency, and solves the practical production problems.

1. Introduction

With the rapid development of computer technology and camera equipment [1], machine vision technology is widely used in for measurement [2], positioning [3] and other fields of industrial production due to its high accuracy and high automation.

Welding technology is an important processing technology in the production process. In industrial production processes, the position of the weld has an important reference for the subsequent production process. In the production process of steel drums, the system needs to perform a series of operations such as screen printing, curling and bottoming on the position of weld seam which has been flattened and painted [4]. The existing welding seam recognition usually adopts the image matching method. The system collects a large number of weld seam images in advance to establish a database. When it works, the captured image is compared with that in the database through feature matching to identify the weld. However, this method is only applicable to continuous welds with large widths. When it comes to the situation that the size of the weld which has been flattened and painted is small, the lines are discontinuous, and the features are not obvious, it is difficult to identify them by image matching. There are also ways to identify welds using laser vision sensor [5] or dark channel [6].

According to the existing literature, this paper proposes a flattened seam recognition technology based on machine vision. We created a machine vision system for welding seam recognition software module for image processing utilize by a GUI application. The system is developed by utilizing...
python language, PyQt5 for graphical interface editing, Modbus protocol for communication between host computer and PLC, and the OpenCV function library for writing image processing functions. The system first performs preprocessing on collected images such as filtering and graying to reduce noise and interference, moreover to reduce the amount of irrelevant image information. In the digital image, the edge detection algorithm is used to extract a clear weld seam image. Finally, the weld seam is extracted by probabilistic Hough transform which can solve the recognition error caused by intermittent welds and achieve the goal of the online recognition of welds seam. To further improve the real-time performance of the system, the system uses a process pool to implement the image processing algorithm. Meanwhile, multi-thread technology is used in the main process to ensure the efficiency of communication and image acquisition.

2. Components of welding seam identification and positioning system

2.1. Overall system scheme
The system consists of a multi-process image processing module(MPIP), an image acquisition module(IA), a multi-threaded communication module(MTC), a PLC motion control module(PLC), and a servo motor motion module(SMM) [7]. The image acquisition module includes an industrial camera and a stable light source.

![Figure 1. System overall structure.](image)

The steel barrel is horizontally placed on the rotating mechanism. The camera collects the surface image of the steel barrel while it is rotating. The image processing module recognizes the image in the process pool. It judges whether the image contains a weld seam according to the probabilistic Hough straight line algorithm, and if so, it calculates the weld position coordinates and sends the value to the motion control module. The PLC motion control module receives the signal from the image processing module and controls the servo motor to stop the welding seam at the expected position.

Multi-process execution can maximize the computing power of the computer and improve the efficiency of program processing. Corresponding to the 4 cores of the CPU, the process pool sets 4 image processing tasks. When one of the processes detects a weld, the remaining tasks are synchronized to stop by the event. Two threads are newly created in the main process for I/O tasks [8], one for Modbus communication between upper and lower computers, the other for image acquisition and transferring to the process pool. The structure of the system is shown in Figure 1.

2.2. System hardware introduction
The system uses the upper and lower computer mode: the upper computer is an industrial PC, and the lower computer is a PLC. The light source is a bar-shaped white LED whose size is 225mmx55mm.
and the camera resolution is 640x480. At this time, the camera frame rate is 30fps, and the exposure time is 2.5ms. The image pre-processing, image segmentation, and welding seam recognition in this system are all calculated in the host computer. In order to conveniently record the test results of the system, the main interface of the system displays parameters such as the width, position, coordinates, and angle of the weld, then marks the weld in the image. The industrial PC communicates with PLC through Modbus protocol. The industrial PC triggers image acquisition and processing according to the signal that the steel barrel is ready which is sent from PLC, and transmits the recognition result to the industrial PC for positioning. The structure of the system is shown in Figure 2.

![Weld recognition control system](image)

**Figure 2.** Weld recognition control system.

### 3. Weld seam recognition algorithm design

#### 3.1. Selection of ROI (regions of interest)

In machine vision, the region of interest refers to the area that needs to be processed from the processed image in the form of boxes, circles, ellipses, irregular polygons, etc. In order to reduce the area of the image that the system needs to process, this system selects a rectangular ROI close to the shape of the weld within the illumination range, which has the following advantages:

1. Significantly reduce the amount of system operations;
2. The background area is reduced; some noise interference is eliminated;
3. Reduce the positioning range of the weld and improve the accuracy of identification and positioning.

The distance between the light source and the steel barrel is 200mm, the width of the light source is 55mm, and the image size of the ROI region is 640x120. It can be known from the calibration that the size of the area of the steel barrel corresponding with image here is 70mm x14mm. Figures 3 to 6 are images of welds at different processing stages.

![Original image of weld seam](image)

**Figure 3.** Original image of weld seam.

![Grayscale image of weld seam](image)

**Figure 4.** Grayscale image of weld seam.

![Gaussian filtered image](image)

**Figure 5.** Gaussian filtered image.

![Canny operator detection image](image)

**Figure 6.** Canny operator detection image.
3.2. Weld image processing

In actual work, the images collected by industrial cameras are mixed with a lot of noise and interference, such as uneven lighting, cracks, surface stains, interrupted welds, and blurred edges. Noise and interference will cause problems such as distortion and deformation of the image, which leads to the image not being used directly for welding seam recognition. Therefore, the system preprocesses the image to reduce noise interference. The pre-processing process includes image graying, image Gaussian filtering, OTSU binarization [10], and Canny operator edge extraction [11].

In welding edge detection, the threshold of the Canny operator needs to be set manually. If the threshold is too high, the edge may be broken or discontinuous. Otherwise, a large number of false edges will be detected. And the optimal thresholds of different images are not consistent, so the edge detection results are unstable. The setting of the fixed threshold cannot meet the requirements of system accuracy. Therefore, the system needs to adopt a method that can automatically apply canny thresholds to achieve system functions. This system uses the method of combining OTSU algorithm and Canny operator, and uses the gray threshold \( T \) obtained by OTSU algorithm as the Canny gradient threshold \( T_2 \) parameter [12]. Image preprocessing code in appendix.

3.3. Weld seam recognition

There are two methods to identify welds based on edge detection. One is to perform a series of morphological processing [13] on the image after image segmentation, the method is as follows:

1. Morphological gradient processing fills the weld seam;
2. Corrosion operation removes small areas of impurities;
3. Expansion treatment to repair broken welds;
4. Identify the welds by extracting the maximum connected domain.

This method has the advantage of rapid weld recognition, but the problem of discontinuity of the weld image caused by the flattening process cannot be solved. In morphological processing, the expansion operation can only be performed in vertical and horizontal directions, which means that only vertical and horizontal welds can be connected, and it fails to link welds with a certain slope. The recognition effect of morphological processing is not satisfactory. Meanwhile, the distance between weld fractures is often large. During the expansion operation, the expansion in a certain direction is increased, which will aggravate the noise interference resulting from incomplete cleaning and affect the recognition result. Simultaneously, Expansion and corrosion operations in morphological processing require special parameters. Too large or too small of this parameter will seriously interfere with the result and cannot meet the general requirements of the system.

The second method treats the weld as a straight line, counts the number of coincident points in the polar coordinate space, and uses the Hough transform line [14] detection method to identify the weld. This method can handle welds with arbitrary slopes, and can identify broken welds, but the standard Hough transform will transform each pixel in the image, which takes a lot of time.

The system uses a detection method based on probabilistic Hough transform to extract welds. The principle of the probabilistic Huff transform is to randomly extract an edge point in the image. If the point has been calibrated as a point on a straight line, continue to randomly select one from the remaining edge points in the connection until all edge points are all drawn. If the point is not marked as a point on a straight line, the point is calibrated; Hough transform is performed on the extracted points after which the accumulation calculation is performed; the point with the largest accumulated value in the Hough space is selected. It is considered as a straight line; the longest line segment is selected from the straight lines, and if the length is greater than the threshold, the end point coordinates are returned as the result of the output. The probabilistic Hough transform code is in the appendix.

The image processing flowchart is shown in Figure 7; Figure 8 is the final recognition result of the system; Figure 9 shows the image processing operation plan.
**Figure 7.** Weld recognition process.

**Figure 8.** Detection results of probabilistic Hough transform.

**Figure 9.** Operation plan.
4. Creation of a weld recognition system

4.1. Overall operation plan

The maximum diameter of the steel barrels studied by the system is $d = 570\text{mm}$. The rotation distance of the barrel after one revolution is $1790.7\text{mm}$. The cycle time of the factory production line is $10\text{s}$. The effective processing width after ROI interception is $14\text{mm}$. Therefore, the steel bucket can be divided into about 128 pieces of processing area. Processing 13 frames per second can guarantee no loss of frames. By adjusting the exposure time, the target processing frame number is set to 30 frames per second. It takes $20\text{ms}$ to run the image processing task once. When 4 identical task image processing processes are used at the same time, real-time requirements can be met. UI response, communication, and image acquisition are in the same process. While ensuring the frame rate, the interface won’t be blocked, which also ensures the real-time communication. The size of the process pool is consistent with the number of CPU physical cores.

4.2. Communication design

4.2.1. Modbus communication. Modbus is a serial communication protocol commonly used in industrial electronic equipment. The system uses a master / slave architecture protocol. The Modbus TCP protocol is used to create a new Modbus communication thread in the main process to synchronize communication. The host computer is the master station, and the PLC of the lower computer is the slave station. When it is working, the host computer reads the signal sent from PLC that the steel barrel is ready, and sends the result signal to the lower computer after recognition. When the host computer is reading data, the identification code is started through the signals and slots mechanism, and the data is written to the PLC by the Queue in the FIFO (First In, First Out).

4.2.2. TcpMaster. Use the module Modbus_tk to implement Modbus communication. The TcpMaster class is located in the modbus_tcp package and is used to build a master object. Port port = 502. Main use method:

```python
master = modbus_tcp.TcpMaster(IPaddress,502)
master.set_timeout (3.0)
execute = (self,slave,function_code,starting_address, quantity_of_x=0, output_value=0, data_format=""", expected_length=-1):
```

Among them, ‘set_timeout’ sets the communication timeout. The ‘execute’ method executes the modbus query and returns data in the form of tuples. The returned tuples depend on the query function code.

4.3. Concurrent design

In the existing mode, it is mostly a single process to capture and process the image. The performance of the multi-core CPU cannot be fully used [15]. If the image processing efficiency is low, there is a possibility that when the program recognizes the weld, the actual position of the weld has turned to the expected position. Therefore, the system uses the parallel computation of the process pool to improve the processing efficiency. The process pool creates multiple processes in advance. After the execution of the task, the process does not disappear, but returns to the process pool to become idle, which saves the overhead of repeated creation and destruction of the process.

The system creates parallel tasks by importing the ProcessPoolExecutor method based on the concurrent.future model, adding tasks and images to the pool through the submit method, using the event event to achieve synchronization, and stopping acquisition and processing functions after recognition.

```python
from concurrent.futures import ProcessPoolExecutor
with ProcessPoolExecutor (max_workers=4) as executor:
    executor.submit (weldreco, [img, event])
```
Among them, ‘weldreco’ is a recognition algorithm function, ‘img’ is a captured image, and ‘event’ is a recognition result event. When each collection is completed, the ‘executor.submit’ method is adopted to join the process pool to operate. In function ‘weldreco’ event is judged whether it has been set in other processes and set according to the result of current process.

5. **Experiment analysis**

5.1. **System hardware configuration and experimental results**

| Table 1. Hardware configuration. |
|----------------------------------|
| OPERATING SYSTEM | MICROSOFT WIN764BIT |
| DEVELOPMENT ENVIRONMENT | python3.7 |
| CPU | Intel Core i5 |
| VIDEO MEMORY | 6G |
| RAM | 8G |
| OPENCV | OpenCV2.4.3 |

| Table 2. Algorithm time statistics: (running 1000 times). |
|-----------------------------------------------------------|
| ALGORITHM MODULE | AVERAGE TIME | AVERAGE TIME (ADDED ROI) |
| GAUSSIAN FILTERING | 9.176ms | 2.728ms |
| OTSU THRESHOLD EXTRACTION | 0.573ms | 0.522ms |
| CANNY OPERATOR EDGE EXTRACTION | 2.230ms | 1.730ms |
| HOUGH TRANSFORM | 18.136ms | 14.994ms |
| PROBABILISTIC HOUGH TRANSFORM | 16.167ms | 13.881ms |

| Table 3. Performance index. |
|-----------------------------|
| SYSTEM INDEX | REQUIREMENT | TIME COST |
| PRODUCTION LINE | 10s | 3.3s |
| FRAME COUNT | 13fps | 30fps |
| IMAGE PROCESSING | 20ms (Single image) | 35ms (Four images) |
| EXPOSURE TIME | 2.5ms | 2.5ms |
“Added ROI” to the table means reducing the size of the initial image to 70mm x 14mm. Multi-process experiments show that it takes 20ms to process one image and 35ms to process four. From the above experimental results in Tables 1 to 3, it can be seen that the use of probabilistic Hough transform and ROI interception can slightly improve the processing efficiency. The overall computing time meets the manufacture rhythm.

6. Conclusions
Weld recognition is an important process in the production process of steel drums. This paper analyzes the development of the existing technology, and proposes a multi-process parallel processing based on machine vision to identify weld seams. The system combines canny edge extraction and probabilistic Hough transform based on OTSU threshold, making full use of the advantages of the multi-core of the host computer to perform parallel processing. Experimental results show that the algorithm can quickly identify the weld position, and meets the processing cycle requirements in real time, which can be used in actual production.

In the future, after accumulating a large number of weld data, the depth neural network (DNN) is applied to weld recognition, which solves the problem of low recognition rate of multilayer neural network such as BP network.

Acknowledgements
Funded by Wuxi Science and Technology Development Fund (Grant No. CGE02G1717); Jiangsu university student innovation and entrepreneurship training program (No. 201910304020Z).

References
[1] Chen Ying 2018 Study on development status and application trends of machine vision technology Wireless Internet Technology 15(19) 147-148
[2] Wu Xiang 2019 Measurement of sizes of non-contact screws based on visual recognition technology Metrology & Measurement Technique 46(08) 40-42
[3] Jiao Rihua 2018 Lane Marking Detection and Lane Positioning Technology Based on Machine Vision Xidian University
[4] Lin Chenbiao 2017 Research and analysis of detection technology in production process of tight head drum Modern Manufacturing Technology and Equipment 10 1-3
[5] Li Keqi, Luan Yizhong, Xia Nan and Guo Zhengliang 2019 Research on weld identification based on industrial robot arc welding Modern Manufacturing Technology and Equipment 10 13-17
[6] Xu Hao, Li Gang and Ma Peisong 2017 Seam image recognition preprocessing based on machine vision Journal of Guangxi University (Natural Science Edition) 42(05) 1693-1700
[7] Cong Huanwu, Guo Fujuan, Lv Fei and Wang Qian 2012 Research on weld seam recognition technology based on CCD image processing Electronic Measurement Technology 35(03) 73-77
[8] Peng Hua 2019 Effect of thread pool scheduling on server performance Communications Technology 52(09) 2172-2176
[9] Tian Yingzhong, Liu Hongfei, Feng Jiecai and Cao Shouqi 2019 Weld width detection method based on vision sensor Metrology & Measurement Technique 46(12) 56-58
[10] Huang Shijie and Zeng Jianhua 2019 Implementation of image edge detection and segmentation based on MATLAB Journal of Chifeng University (Natural Science Edition) 35(09) 47-49
[11] Yang Xianfeng , Wu Yuanyuan and Zhao Ling 2018 Pipeline crack detection algorithm based on Canny detector Computer Engineering and Design 39(03) 798-803
[12] Liang Zhaojun and Zhong Jun 2019 Adaptive Canny algorithm improvement based on Otsu algorithm and histogram analysis Modern Electronics Technique 42(11) 54-58
[13] Wang Xiang and Luo Suyun 2019 Lane detection based on Otsu threshold and morphology Agricultural Equipment & Vehicle Engineering 57(10) 69-72
Appendices

Image preprocessing code
   
   ```
   gray = cv2.cvtColor (img, cv2.COLOR_BGR2GRAY)
   gauss = cv2.GaussianBlur (gray, (29,1), 10)
   ret, dst = cv2.threshold (gauss, 0,255, cv2.THRESH_BINARY + cv2.THRESH_OTSU)
   edges = cv2.Canny (gauss, ret / 2, ret, apertureSize = 3)
   ```
   
   Where ‘img’ is the original image of the weld, ‘cv2.COLOR_BGR2GRAY’ refers to the color space conversion, ‘gray’ is the grayscale image, ‘gauss’ is the Gaussian filtered image, ‘(29,1)’ is the Gaussian matrix, ‘10’ is the Gaussian filter standard deviation, and ‘ret’ is the OTSU algorithm return ‘dst’ is the binarized image, ‘0,255’ is the binarized pixel range, ‘cv2.THRESH_BINARY + cv2.THRESH_OTSU’ divides the pixel values of the foreground and background pixels, and ‘edges’ are the images extracted by the canny operator.

The probabilistic Hough transform code:
   
   ```
   lines=cv2.HoughLinesP (image,rho,theta,threshold,minLineLength,maxLineGap)
   ```
   
   The ‘image’ in the code is the image after canny edge extraction; ‘rho’ is the polar angle step size in pixel units, and the accuracy is in pixel units; ‘theta’ is the polar step size precision in radians, here is taken as ‘π/180’; ‘threshold’ is the minimum number of intersections required for a straight line in Hough space. Multiple line segments greater than the threshold are considered to be on a straight line, here is taken as 300; ‘minLineLength’ is the minimum number of pixel points that make up a line segment, if less than this, the line segment will be truncated, here is taken as 50; ‘maxLineGap’ is the maximum distance between two line segments on a straight line, here is taken as 20.