Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **2**, March **2016**

A Novel Algorithm of Rebar Counting on Conveyor Belt Based on Machine Vision

Zuoxian Nie[1,a,b], Mao-Hsiung Hung*[,a], Jing Huang[2,a]

[a]College of Information Science and Engineering, Fujian University of Technology
[b]Fujian Provincial Key Laboratory of Digital Equipment, Fujian University of Technology
No.3, Xueyuan Road, University Town, Minhou, Fuzhou City, 350118, China
*Corresponding author, mhhung@fjut.edu.cn
[1]:zxnie@qq.com, [2]:419467569@qq.com

Received July, 2015; revised December, 2015

Abstract. *Automatic counting system plays an important role to accomplish the bundling of fixed amount in the rebar production line. In this paper, we present a novel algorithm that automatically counts rebar on production line based on video analysis. Our algorithm allows that conveyor belt moves at rather higher speed and rebars roll forward or backward on conveyor belt. There are mainly two parts in our proposed algorithm. Firstly, a method is developed to divide the video frame into three partitions, and then a set of heuristic rules are deduced based on these partitions. Secondly, for cases excluded by above rules, a sequence matching algorithm is developed to identify repeated sequence of rebar between adjacent frames, and then number of newly arrived rebar in every frame can be computed. The experimental results show that our method can support high-speed and precise rebar counting.*
Keywords: Automatic counting system, Conveyor belt, Rebar production.

1. **Introduction.** On a rebar production line, rebar is cut to a certain length and transmitted to counting and bundling station by a conveyor belt. At the counting station, rebar is counted continuously. When a preset number of rebars pass by the counting station, for example, 100, the conveyor belt needs to be halted temporarily, and then workers can bundle the rebar together and remove them from the conveyor belt. After that, the conveyor belt resumes to run and the counting restarts. Traditional manually counting rebar is a very heavy and monotonous task, and inaccurate counting results often happen due to long-time works. Therefore, to improve production effectivity, replacing manually counting with automatic rebar counting becomes very important.

Image analysis based object counting technologies are already widely used in transportation management, agriculture and industrial production. To count stacked corrugated boards, Ufuk et. al. [1] apply image operation such as sober filtering, erosion, dilation, mean filtering and thresholding to separate corrugated boards from each other, and then its easy to find number of corrugated boards by counting separated objects in these images. To count small soft fibrils, Xiao et. al. [2] proposed to get the optimal threshold based on SOFM neural network, and then perform image segmentation, boundary separation and object counting. Detection of prominent pest attacks at early stage is a important task in greenhouse crops production, Ikhlef et. al. [3] proposed a pattern recognition algorithm to extract the locations of harmful insects and count them from the surveillance video. In above works, the objects are static, however, it is more challenging to count moving objects based on image processing. Currently, counting vehicles on highways and in cities is a hot topics [4], the main approaches are to identify and track vehicles between consecutive frames by extract different features such as SIFT [5] and HOG [6] from video frames. These approaches are not suitable for rebar counting because most cut ends of rebars are round and the extracted features will be too similar to distinguish.

One of the simplest method of automatic rebar counting system is photoelectric sensor technology [7]. When a rebar on conveyor belt passes by the photoelectric sensor to trigger a electric pulse, the counter accumulates one accordingly. However, the method has met a serious problems. A rebar has chances to adhere to another or pile above another, which makes it difficult for photoelectric sensor technology to count rebars correctly. To avoid the cases of adhesion and piling, a separating mechanism is required to install before the counting system, to make every rebar apart from each other. However, the separating mechanism often slows down the speed of conveyor belt to limit production.

Several systems of automatic rebar counting based on still images have been proposed [8]-[13]. The images of a bundle of rebars are captured, and then machine vision software segment the rebar objects in the images to obtain the counting results. In the practical production, gray-level changes caused by oxidation on the rebar end, and mutual occlusion of rebar objects make it difficult to correctly segment rebar objects from background. Zhang et. al. [8] use weighted mean filtering to reduce noise and morphological open operation to remove the disturbance of rebar body, and then use round template to match rebar. However, the counting based on still images only obtains the object numbers of rebar bundles. It cannot perform a preset number of rebar bundling according to the counting result.

Therefore, instead of still image, the automatic counting based on videos becomes the most popular techniques in rebar production line [14]-[17]. The videos of rebar running on the conveyor belt are captured by a camera, and then machine vision software analyzes the image sequences to obtain counting results. The advantages of video-based counting system includes: 1) no modification of conveyor belt for installing the counting system, 2) ability to accomplish a preset number of bundle, and 3) no effect of the running speed of conveyor belt due to the counting.

Luo et. al. [15] have proposed on-line counting system using machine vision for steel bar production. In order to recognize repeated rebar sequence in adjacent two frames, the algorithm computes amount of movement of rebar relating to their counterpart in the previous frame. A detection window is set at center of every frame, and once some rebar at the detection window that cannot be matched to rebar in the detection window in previous frame, they are regarded as new arrivals and counted.

To solve the problem of multiple moving targets tracking in the online rebar counting system, Chen and Li [17] proposes a fast and stable online rebar tracking algorithm, which fuses the characteristics of movement, geometry and location in the vertical projection curve derived from the binary image of the rebar's end. The algorithm can adaptively select the position and size of the tracking window, and evaluate a confidence value which combines the distance correlation and area similarity of the rebar objects in adjacent two frames. In addition, it depends on the movement estimation of conveyor belt to match rebar between adjacent two frames. Rebar rolling caused by the acceleration and deceleration of conveyor belt is also considered in the movement estimation. As a result, the algorithm allows the tracking process better fault tolerance.

Su et. al. [14] have designed a capturing equipment with two-light illuminating in different directions to improve the acquirement quality when inter-covering of rebars happens. In addition, their proposed system reduces background luminance close to zero and enhances rebar luminance close to saturation using an elaborate setting of a high power plant-lights and cameras exposure time. For the purpose of tracking rebar, the centers of them are extracted by regional-max operator. Subsequently, vertical projection and minimal square difference template matching is used to track repeated rebar between frames. Finally, a single vertical line is used to count rebar.

In counting systems based on video analysis, a detection window is often set in frame images, when a rebar occurs in the detection window, it is counted. However, when a rebar moves fast, it may occur before the detection window in the previous frame and after the detection window in the current frame, so it will not appear in the detection window, and the system will fail to count it.

In our system, no detection window is used, and rebars are counted as soon as it appears in a frame image. This way brings three advantages as follow.
(1) The conveyor belt can run at faster speed without the limitation of the detection window.
(2) There is more time to smoothly slow down the conveyor belt when the counting reaches the preset number. As a result, rebars rolling caused by sharp braking can be greatly reduced.
(3) The image of the last counting rebar certainly stays in the video frame, when the conveyor belt stops. Therefore, the packing worker easily identify which rebar is the last one from the monitoring screen.

There are mainly two parts in our proposed algorithm. Firstly, a method is developed to divide the video frame into three partitions, and then a set of heuristic rules are deduced based on these partitions. Secondly, for cases excluded by above rules, a sequence matching algorithm is developed to identify repeated sequence of rebar between adjacent frames, and then number of newly arrived rebar in every frame can be computed. The experimental results show that our method can support high-speed and precise rebar counting.

The remaining sections of this paper is organized as follows. Section 2 describes the proposed method, which includes image segmentation, counting algorithm and sequence matching. In Section 3, we first describe the hardware architecture, and then demonstrate the performance evaluation of the proposed method. The conclusions are drawn in Section 4.

2. **Proposed method.**

2.1. **Image Segmentation.** The first processing of the proposed method is to segment rebar objects from each frame. There are many existing works that focus on segmenting rebar objects from various tough cases of rebar images. Actually, on a rebar production line, proper illumination and image acquirement can make it simpler. From observation of the capture images, we learned that the rebar objects are regions with high gray level, and the region areas are within a fixed size range. Therefore, a simple method of fixed thresholding and region selection, is feasible to segment objects from original frames. We first draw a rectangle above the conveyor belt as a ROI (region of interest). When rebar objects are carried forward by the conveyor belt, the rebar objects always runs through the ROI in the sequential frames. Fig.1(a) displays a frame of rebar objects running on the conveyor belt. The round regions of high gray level are rebar objects and the lower large region of high gray level is the conveyor belt.

First at all, we reduce the domain by the ROI in a frame. In the domain, a fixed thresholding is used to segment out a region (R)of higher gray level , as

$$p \in R, \text{if } g_{\min} \leq g \leq 255 \tag{1}$$

where a pixel (p) has a gray level (g) and $g_{\min}$ is a threshold. Although there are many existing algorithms to find optimal threshold of images [18], but in our system, the background of video frames and luminance is are under well-controlled conditions, so to save some computational cost, it is reasonable to choose a fixed threshold. According to our experiments, intensities of pixels in background images range from 80 to 180, and most pixel intensities fall into the interval of 80 to 120. By contrast, pixels within cut ends of rebars have apparent higher intensities than pixels of the background. After experimental analysis, we set the threshold to e.g. 180 for our system. Then, the R can be labeled several connected regions (R_{conn}) based on 8-connective rule. Among R_{conn}, we select the regions whose areas are within a fixed size range of $[A_{\min}, A_{\max}]$, as

$$R_{\text{sel}} = \arg_{r \in R_{\text{conn}}} \left(A_{\min} \leq area(r) \leq A_{\max} \right) \tag{2}$$

where R_{sel} means the selected regions and $area(.)$ return the areas of regions. $A_{\min}$ and $A_{\max}$ respectively represent a smallest size and a largest size to select rebar objects from the connected labeled regions. As a result, the regions of the rebar objects are segmented out, as shown in Fig.1(b). The segmentation method is quite simple but effective to extract rebar objects from each frame.

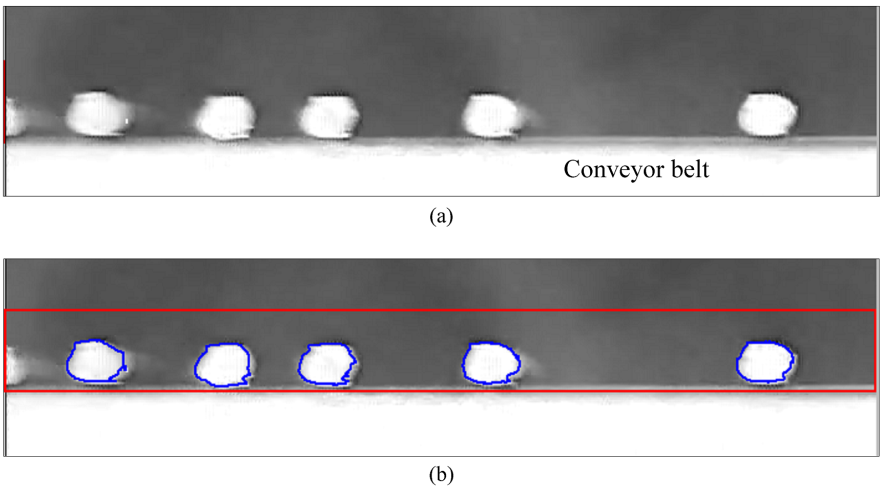


Figure 1. Image segmentation: (a) Original frame, (b) ROI of red box and segmentation result

2.2. **Counting Algorithm.** Using the segmentation results of rebar objects, our proposed counting algorithm performs frame by frame. First of all, we partition the ROI of rebar objects into three areas including entrance, buffer and exit areas. The entrance area locates in the left part of ROI in which an object possibly appears at the first time in the video sequence. The exit area locates in the right part of ROI in which a object possible appears at the last time in the video sequence. The buffer area means the area that excludes entrance and exit areas from ROI. Fig.2 shows three partitions of the ROI for object counting.

The width setting of the entry and exit areas should refer the longest distance which a object run at a frame time. Assumed that images are captured with width w (in

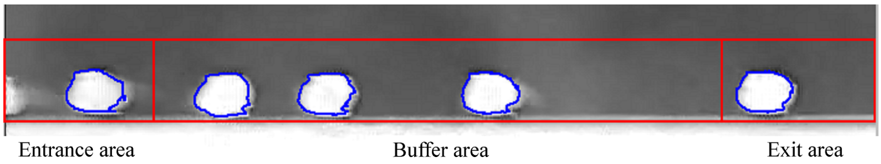


Figure 2. Three partitions of ROI for object counting

pixel), corresponding to actual width w_a (in cm) of conveyor belt. If the maximum speed of conveyor belt is s (in cm/sec), and image is captured every t (in sec). Therefore, maximum distance D (in pixel) which a rebar object can move forward at adjacent two frames, can be computed as

$$D = s \cdot t \cdot (w/w_a) \tag{3}$$

For example, assumed that images are captured with width of 600 pixels, corresponding to actual width 50 cm of conveyor belt. If the maximum speed of conveyor belt is 60 cm/s, and images are captured at 25 fps, then maximum distance one rebar object can move forward at adjacent two frames will be $D = 60(1/25)(600/50) = 28.8$ pixels. Therefore, width of entrance and exit area can be set as 29 pixels. Entrance and exit area can be adjusted wider to allow higher speed of conveyor belt.

In our proposed method, we accumulate rebar objects frame by frame, i.e. the counting result of the current frame is equal to the counting result of the previous frame plus the number of newly-arrived objects in current frame, as

$$Cnt(k) = Cnt(k-1) + N_{new} \tag{4}$$

where $Cnt(k)$ and $Cnt(k-1)$ are counting results in the current and previous frames respectively, and N_{new} is the number of new-arrived objects in the current frame.

According to the definition of width of entrance and exit area, some useful knowledge can be drawn. Newly-arrived rebar objects in a frame must occur in its entrance area, and its possible only for rebar objects in exit area to disappear from next frame. Rebar objects in exit area of current frame may come from buffer or exit areas at previous frame, but rebar objects in exit area of previous frame may already disappear from current frame. Rebar objects in entrance area of current frame may be newly-arrived objects or come from entrance area of previous frame.

In the counting algorithm, we first apply four heuristic rules to determine the number of newly-arrived objects. These rules use the relationships of the object number detected in entrance, buffer and exit areas from the current and previous frames. The four rules are described in the following.

R1. When no object is in the entrance area in the current frame, Nnew is equal to zero.
R2. When no object is in the entrance area in the previous frame, the object(s) in the entrance area is regarded as newly-arrived object(s).
R3. No object appears in the exit area in the previous frame. It means that all of object(s) the previous frame still appears in the current frame. Thus, N_{new} is equal to the difference of number of rebar objects between the current frame and the previous frame.
R4. In the current frame, no object is in the exit area, but there are object(s) in exit area of the previous frame. It means that object(s) in the entrance and buffer areas in the previous frame, still appears at the current frame. Thus, N_{new} is equal to the difference from the object number of the current frame to the object number of the entrance and buffer areas at the previous frame.

The proposed counting algorithm states in pseudo code as follow, where k-th frame means the current frame and $(k-1)$-th frame means the previous frame.

Counting Algorithm
// Variable definition
$N_{entr}(k)$: number of objects in entrance area at k-th frame
$N_{buff}(k)$: number of objects in buffer area at k-th frame
$N_{exit}(k)$: number of objects in exit area at k-th frame
$N_{entr}(k-1)$: number of objects in entrance area at $(k-1)$-th frame
$N_{buff}(k-1)$: number of objects in buffer area at $(k-1)$-th frame
$N_{exit}(k-1)$: number of objects in exit area at $(k-1)$-th frame
$N(k)$: total number of objects at k-th frame
$N(k-1)$: total number of objects at $(k-1)$-th frame
N_{new}: number of newly-arrived objects from $(k-1)$-th to k-th frame
N_{rep}: number of repetitive objects from $(k-1)$-th to k-th frame
$Cnt(k)$: counting result at k-th frame
$Cnt(k-1)$: counting result at $(k-1)$-th frame

Begin
  if $N_{entr}(k)$=0 then // R1
    $N_{new} = 0$
  else
  if $N_{entr}(k-1) = 0$ and $N_{entr}(k) > 0$ then // R2
    $N_{new} = N_{entr}(k)$
  else
  if $N_{exit}(k-1) = 0$ and $N_{entr}(k) > 0$ then // R3
  else
  if $N_{exit}(k-1) > 0$ and $N_{exit}(k) = 0$ then // R4
    $N_{new} = N(k) - (N_{entr}(k-1) + N_{buff}(k-1))$
  else
  {
    // if none of the four rules is selected, then call fnNumRepObj(.) to obtain N_{rep}
    N_{rep}= fnRepNumObj($N(k-1)$,$N(k)$,$N_{entr}(k)$, $N_{exit}(k-1)$, S_a, S_b, θ)
    N_{new}= $N(k)$-N_{rep}
  }
  $Cnt(k)$=$Cnt(k-1)$+N_{new} // update the counting result
End

Fig.3 demonstrates the four examples to illustrates the four rules. In each sub-figure, the upper and lower drawings respectively represent the previous frame and the current frame. Each frame is partitioned into entrance, buffer and exit areas from left to right. As shown in Fig.3(a), according to R1, $N_{entr}(k) = 0$, so $N_{new} = 0$. As shown in Fig.3(b), according to R2, $N_{entr}(k-1) = 0$ and $N_{entr}(k) > 0$, so $N_{new} = N_{entr}(k) = 1$. As shown in Fig.3(c), according to R3, $N_{exit}(k-1) = 0$ and $N_{entr}(k) > 0$, so $N_{new} = N(k)-N(k-1) = 8-6 = 2$. As shown in Fig.3(d), according to R4, $N_{exit}(k-1) > 0$ and $N_{exit}(k) = 0$, so $N_{new} = N(k) - (N_{entr}(k-1) + N_{buff}(k-1)) = 6 - (4 + 1) = 1$.

2.3. **Sequence Matching.** Rejected by the above mentioned four rules, these counting cases are required to match objects of the two consecutive frames to obtain the number of newly-arrived objects. Thus, we define N_rep as the number of repetitive objects from $(k-1)$-th to k-th frame, and the number of newly-arrived objects in k-th frame is equal

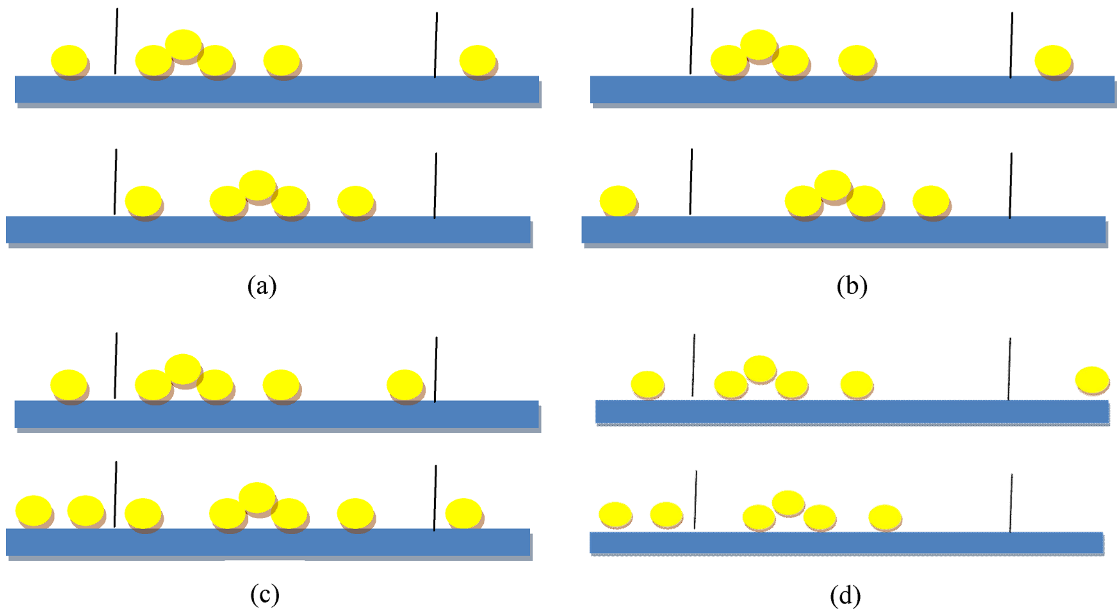


FIGURE 3. Illustration of four rules: (a) R1, (b) R2, (c) R3, (d) R4

to $N(k) - N_{rep}$. The repetitive object means the object appears at both of previous and current frames. Given the x-coordinates of objects in the previous and current frame, we respectively obtain two coordinate sequences of $S_a = (a_i)$ and $S_b = (b_i)$ where ai and bi means a x-coordinate of a object's center. After that, we applied the sequence matching to determine the number of repetitive objects from $(k-1)$-th to k-th frame. As shown in Fig.4, the subsequence of $(a_1, a_2, ..., a_6)$ of S_a exactly matches the subsequence of $(b_2, b_3, ..., b_7)$ of S_b to learn six repetitive objects in two consecutive frames.

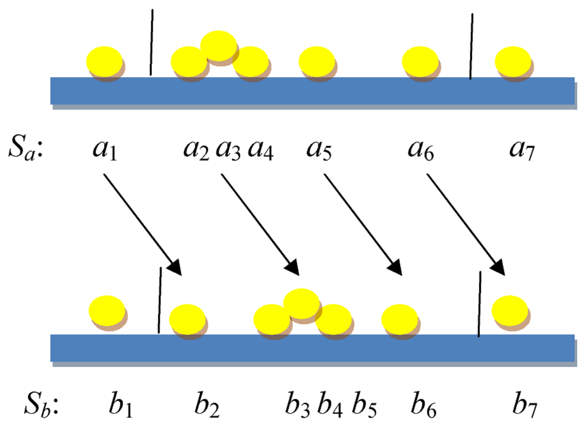


FIGURE 4. Sequence matching from previous to current frame

Before sequence matching between the two consecutive frames, the exact length of the sequence of the repetitive objects is unknown. Thus, we define a lower bound ($Lower$) and a upper bound ($Upper$) of the possible length of the sequence, and a variable (m) increasing from Lower+1 to Upper by step 1. Then, first m-long subsequence of S_a (object sequence at $(k-1)$-th frame) and last m-long subsequence of S_b (object sequence at k-th frame) are extracted.

After the subsequence extraction, we find maximum shift among any two objects in the two subsequence. For example, c_i and c_j are two objects of in m-long subsequence S_c at $(k-1)$-th frame for $i \neq j$, and d_i and d_j are two objects in m-long subsequence S_d at k-th frame for $i \neq j$. If two subsequences are considered as same objects at two adjacent frames, on the condition without rebar rolling, the distance between c_i and c_j should be equal to the distance between d_i and d_j. Thus, when the two distances are not the same, it indicates that the rebar rolling happens, and the absolute difference of the distance is considered as the shift caused by rebar rolling. If the shift of any two objects in the subsequences is less than a threshold (θ), the two m-long subsequences of the previous and current frames are still considered matched. In other words, the examination verifies that there are at least m repetitive objects from $(k-1)$-th to k-th frame, so that the number of repetitive objects updates to be m. In addition, the θ threshold can be defined as the largest shift of two objects from $(k-1)$-th to k-th frame, and it is preset experimentally.

The function to find the number of the repetitive objects (fnRepNumObj) states in pseudo code as follow.

Function: N_{rep}= fnRepNumObj($N(k-1)$,$N(k)$, $N_{entr}(k)$, $N_{exit}(k-1)$, S_a, S_b, θ)
/* Input: $S_a = (a_i)$ for $i = 1, 2, ..., N(k-1)$ and S_a represents a sequence containing x-coordinates (a_i) of objects in the previous frame. $S_b = (b_i)$ for $i = 1, 2, ..., N(k)$ and S_b represents a sequence containing x-coordinates (b_i) of objects in the current frame. θ represents the largest shift of the coordinate interval of two objects from $(k-1)$-th to k-th frame */

$Lower = \max(N(k-1) - N_{exit}(k-1), N(k) - N_{entr}(k))$
// Lower represents the minimum of repetitive object from $(k-1)$-th to k-th frame
$Upper = \min(N(k-1), N(k))$
// Upper represents the maximum of repetitive object from $(k-1)$-th to k-th frame

$N_{rep} = Lower$
// N_{rep} is initialized by Lower
for $m = Lower + 1$ to $Upper$ step by 1
// examine every subsequence whose length varies from $Lower + 1$ to $Upper$
  $S_c = (c_1, c_2, ..., c_m) = (a_1, a_2, ..., a_m)$
  // S_c represents a subsequence of first m objects at the previous frame
  $S_d = (d_1, d_2, ..., d_{m-1}, d_m) = (b_{N(k)-m+1}, b_{N(k)-m+2}, ..., b_{N(k)-1}, b_{N(k)})$
  // S_d represents a subsequence of last m objects at the current frame

  // find maximum shift among any two objects in the two subsequence S_c and S_d
  $maxShift = 0$
  for i=1 to $m-1$ step by 1
    for $j = i + 1$ to m step by 1
      $ad_1 = abs(c_i - c_j)$
      $ad_2 = abs(d_i - d_j)$
      $shift = abs(ad_1 - ad_2)$
      if $shift > maxShift$ then $maxShift = shift$
    endfor
  endfor

  if $maxShift \leq \theta$, then $N_{rep} = m$

// when $maxShift$ is less than θ, N_{rep} updates to be m endfor
return N_{rep}

3. **Experimental Results.**

3.1. **Hardware architecture.** The hardware architecture of our proposed counting system is shown in Fig.5. It consists of an industrial PC, a camera, a LED light and a I/O communication broad. The LED light provides illumination on the conveyor belt. When the rebar objects run through the lighting zone of the conveyor belt, the camera can capture the images of the rebar's section and send the images to PC host. The machine vision software performs the counting algorithm to obtain the number of rebar objects carried by the conveyor belt. In our proposed system, users can preset a packing number. When the counting number reaches the preset number, PC controls PLC to stop the running conveyor belt, so that the packing machine can pack a preset amount of rebar. After the packing finishes, the conveyor belt resume to run and the counting restarts. Fig.6 displays four real pictures of the equipment.

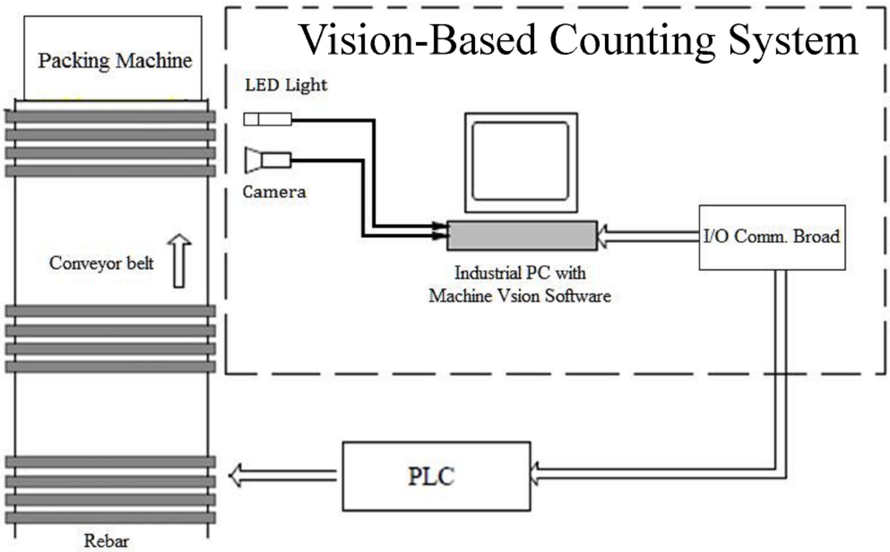
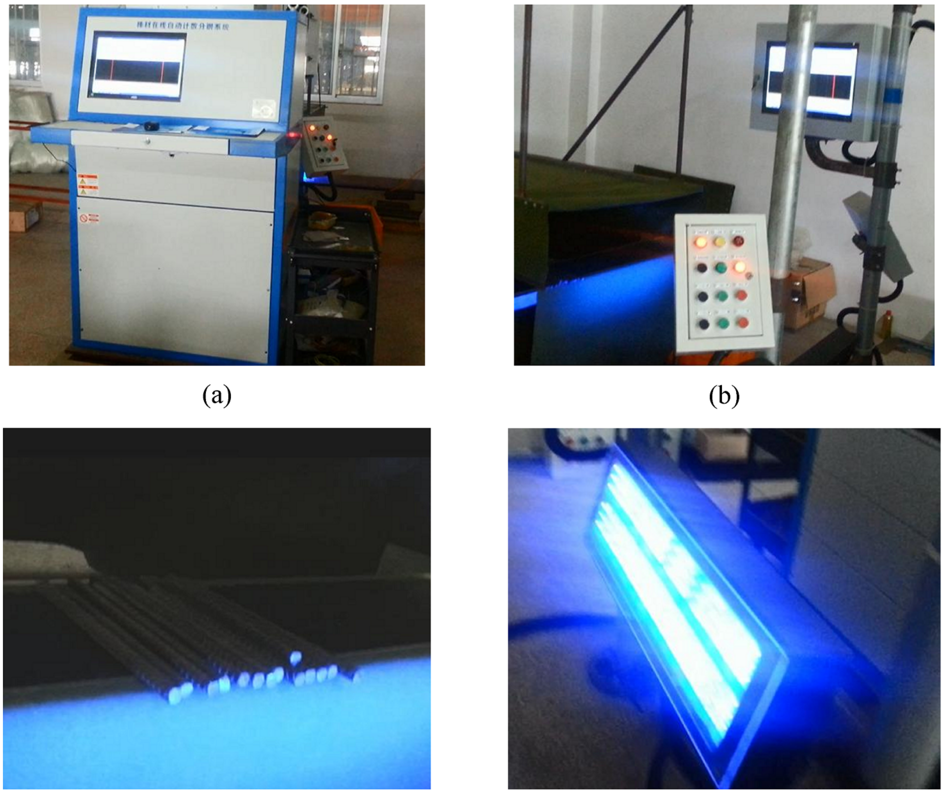


FIGURE 5. Hardware architecture of rebar counting system

3.2. **Evaluation.** To evaluate the performance, we input a video of rebar objects running on the conveyor belt to test the proposed method. The frame rate of the video is 30 fps, and the length is 334 frames. There are 14 rebar objects running through the video sequence. The occurrences of rebar objects are very random, and the intervals of neighbor two objects change a lot. Fig.7 demonstrate the counting results for the 14 rebar objects. Each subfigure displays the total count at the captured frame, and every rebar object is labelled its number. It notes that when no.7 object appeared at 110-th frame, the no.1-no.6 objects have run out of the image frame. As same, when no.10 object appeared at 183-th frame, the no.7-no.8 objects have run out of the image frame, and when no.12 object appeared at 235-th frame, the no.10-no.11 objects have run out of the image frame. In addition, the figure displays that once an object enters the image frames, the counting



(a) (b)

(c) (d)

Figure 6. Real pictures of equipment: (a) Computer host, (b) Control panel and monitor, (c) Conveyor belt and rebar, (d) LED light

number immediately increases. That result indicates that our proposed method is quite effective.

In the 334 frames, we found that there are only 7 frames needing to call the sequence matching algorithm to determine counting results. The observation indicates that the heuristic rules perform the significant role in the proposed method and greatly reduce the execution times of the sequence matching. That makes the proposed algorithm much more efficient than a counting method only using sequence matching.

3.3. **Computational comparison.** To obtain a comparison between the traditional method and our proposed method, we calculate the computational costs of them. We conduct a traditional method of a detection window to count rebar objects, as shown in Fig.8. The method uses the projection of gray level to compare with a preset template of a target object [19]. The counting window locates at the center of the image frame which is drawn in blue. A red signal in the below is the vertical projection of the gray-level of the frame image. When the wave of the projection signal passes the window and compares well with a template of vertical projection of a rebar object, the object counter automatically increase 1. A video of 334 frames containing 14 rebar objects inputs and evaluates the two method. Table 1 lists their processing time. Although both of methods counted rebar objects correctly, our proposed method is more efficient than detection window in the computational cost.

Table 1. Computational comparison

	Detection window	Proposed
Processing time	40.5431 sec	12.9457 sec

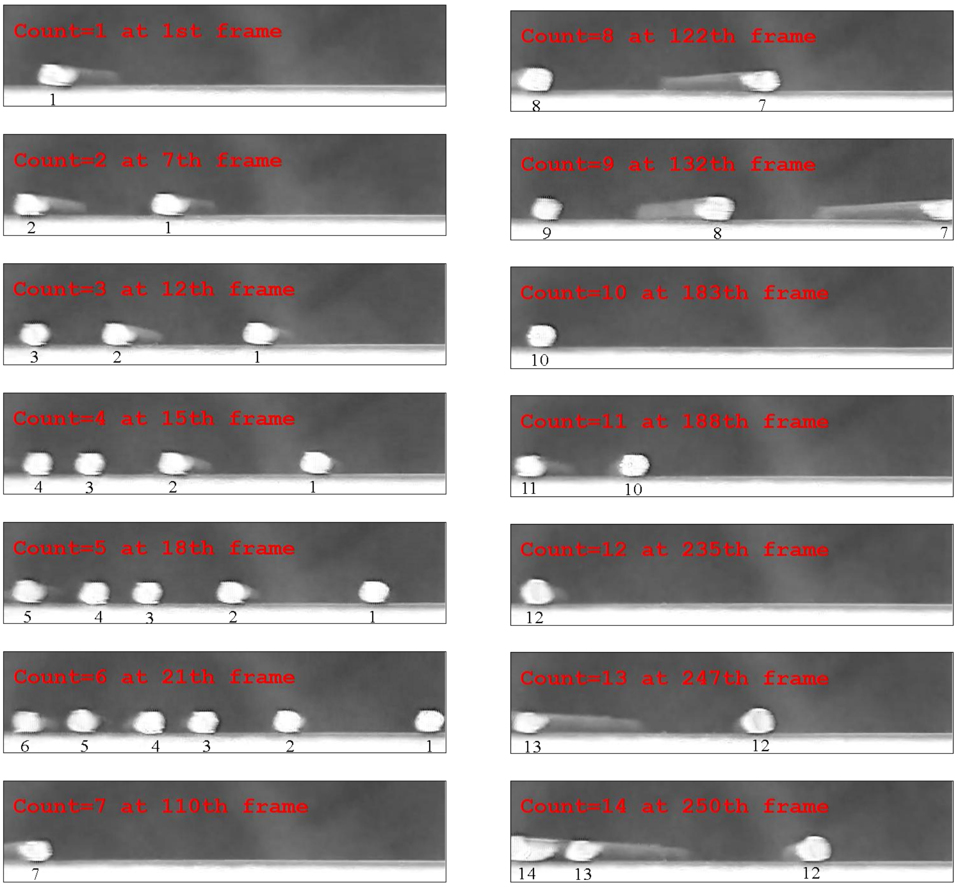


Figure 7. Counting result

4. **Conclusions.** This paper have presented a machine vision algorithm to count rebar on conveyor belt. Not to influence the speed of the conveyor belt, we develop a counting algorithm combining heuristic rules and sequence matching. Based on the proposed partition of three areas, the heuristic rules and sequence matching perform effectively and efficiently for rebar counting. The experimental results indicate that the counting algorithm is very accurate and promising. In addition, the proposed algorithm has the advantages such as smoothly braking and easily monitoring for bundling rebar.

Acknowledgment. This work is supported in part by major platform project of science and technology in Fujian Province (No.2014H2002), major science and technology projects in Fujian Province (No.2014HZ0004-3), provincial university foundation in Fujian Province (No.JK2013030), educational youth key foundation in Fujian province

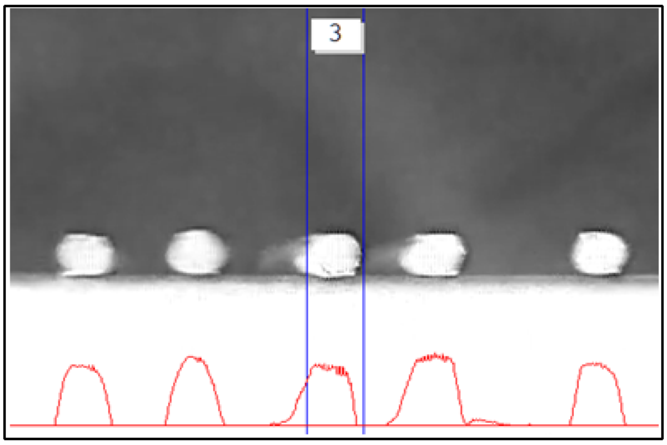


FIGURE 8. Detection window counting rebar objects

(No.JA13210), and research project of Fujian Provincial Department of Education (No. JB11115).

REFERENCES

[1] U Cebeci, F Aslan, M elikH Aydn, Developing a New Counting Approach for the Corrugated Boards and its Industrial Application by Using Image Processing Algorithm, *Practical Applications of Intelligent Systems*, pp. 1021-1040, 2004.
[2] X. X. Xing, L. Fu, and W. W. Shang, Research on an automatic counting system of soft fibrils collection, *2nd International Conference on Computer Science and Network Technology (ICCSNT)*, 2012.
[3] F. Bremond, M. Thonnat, S. Moisan, I Bechar, On-line video recognition and counting of harmful insects, *20th International Conference on Pattern Recognition (ICPR)*, 2010.
[4] M. Liang, X. Huang, C. Chen, X. Chen, A. Tokuta, Counting and Classification of Highway Vehicles by Regression Analysis, *IEEE Trans. on Intelligent Transportation Systems*, vol.16, no.5, pp.2878-2888, 2015.
[5] T. Gao, Z. Liu, W. Gao, J. Zhang, Moving vehicle tracking based on SIFT active particle choosing, *Advances in Neuro-Information Processing*, Springer Berlin Heidelberg, pp. 695-702, 2009.
[6] N. Buch, J. Orwell, S. A. Velastin, 3D extended histogram of oriented gradients (3DHOG) for classification of road users in urban scenes, *British Machine Vision Conference BMVC 2009*, London, Sept. 2009.
[7] Z. Jiayan Z, J. Jin, Study of automatic counting in on-line production of bar steel, *Journal of Huazhong University of Science & Technology*, vol. 32, no. 9, pp.79-81 2004 (in Chinese).
[8] D. Zhang, Z. Xie Z, C. Wang, Bar section image enhancement and positioning method in on-line steel bar counting and automatic separating system, *Image and Signal Processing, 2008. CISP'08. Congress on. IEEE*, vol. 2, pp. 319-323, 2008.
[9] S. Luo, F. Xiao, A new method of recognizing quasi-circular object, *Journal of Central South University of Technology*, vol. 5, no. 4, pp. 632-637 2004 (in Chinese).
[10] S. Lou, J. Huang, Y. Li, Method for Steel Bar Recognizing and Counting based on Multi-camera vision Fusion, *Computer Engineering*, vol. 34, pp. 231-233 2008(in Chinese).
[11] X. Liu, W. Bing, Z. Su, J. Guo, K. Fang Method for Automatic Counting of Steel Bars Based on Mathematical Morphology *Control Engineering of China*, vol. 20, no. 2, pp. 286-285, 2013(in Chinese).
[12] D. Zhang, Z. Xie, J. Ai Research on rebar section positioning method in online rebar counting system, *Chinese Journal of Scientific Instrument*, 2010 (in Chinese).
[13] W. Hou, L. Zhang, M. Dang, Z. Duan, Y. Song, H. Wang, Design and implementation of bar count measurement system based on image processing, *Chinese Journal of Scientific Instrument*, vol. 34, no. 5, pp. 1100-1106, 2013(in Chinese).

[14] Z. Su, K. Fang, Z. Peng, Z. Feng, Rebar automatically counting on the product line, *Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on*, vol. 2, pp. 756-760, 2010.

[15] S. Luo, S. Sha, On-line Visual Counting System for Production of Steel Bar, *Minimicro Systems*, vol. 25, pp. 671-675 2004 (in Chinese).

[16] D. Wang, Q. Sun, Y. Y. Shen, Y. Wang, Y. Li, Development of the automatic counting system for steel bar, *Manufacturing Automation*, vol. 14, pp. 019, 2011(in Chinese).

[17] W. Chen, J. Li, Online rebars tracking algorithm based on multi-feature fusion, *Journal of Fujian University of Technology* vol. 6, pp. 517-521, 2013 (in Chinese).

[18] J. Sauvola, M. Pietikinen, Adaptive document image binarization, *Pattern Recognition*, vol.33, no.2, pp. 225-236, February 2000.

[19] M.-H. Hung, J.-S. Pan, A Real-time Action Detection System for Surveillance Videos Using Template Matching, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 6, pp. 1088-1099, Nov. 2015.